DESARROLLO DE UN PROTOTIPO PARA LA GESTIÓN DE RIESGO CREDITICIO UTILIZANDO TÉCNICAS DE CLASIFICACIÓN SUPERVISADA

DESARROLLO DE SOFTWARE

DIEGO ANDRÉS BORRERO TIGREROS

Código: 201629069

E-mail: diego.borrero@correounivalle.edu.co

Director:

OSCAR FERNANDO BEDOYA LEYVA, Ph.D.

Profesor de la Escuela de Ingeniería de Sistemas y Computación

E-mail: oscar.bedoya@correounivalle.edu.co

Universidad del Valle

Facultad de Ingeniería

Escuela de Ingeniería de Sistemas y Computación

Programa Académico de Ingeniería de Sistemas

Tabla de contenido

Listado de tablas

Listado de figuras

Resumen

En el presente trabajo de grado se desarrolla un prototipo para la gestión de riesgo crediticio utilizando técnicas de aprendizaje supervisado. Este prototipo se propone como un apoyo para el área de gestión de riesgo y tiene como objetivo identificar clientes que puedan incurrir en un estado de mora generando un posible riesgo de crédito para las entidades financieras. En particular, se proponen modelos basados en tres técnicas de aprendizaje supervisado (redes neuronales, árboles de decisión y máquinas de soporte vectorial) para predecir el próximo pago de la cuota de un cliente a partir de datos básicos de la operación, del cliente y de pagos de cuotas anteriores registradas. Las tasas de precisión alcanzadas por los modelos propuestos en los conjuntos de prueba están entre el 60.32% y 78.31%, siendo la técnica de árboles de decisión más precisa que las demás estrategias utilizadas.

CAPÍTULO 1

INTRODUCCIÓN

Las compañías de financiamiento comercial son aquellas entidades que tienen como función principal captar recursos a término con el objeto de realizar operaciones activas de crédito para facilitar la comercialización de bienes y servicios (Superintendencia Financiera de Colombia, 2017). Para las compañías que ofrecen servicios de crédito financiero es muy importante tener una liquidez financiera que permita cumplir con todas las obligaciones adquiridas y tener un flujo de efectivo constante para continuar prestando el servicio, como también disminuir el riesgo de crédito por ser una característica propia a la actividad que desarrollan. Para esto, es importante realizar procesos y análisis que permitan disminuir el riesgo de liquidez y crédito ya que son las principales incertidumbres a las cuales se enfrentan. Poder detectar clientes con insolvencia crediticia antes de que incumplan con sus obligaciones contractuales es un desafío para estas entidades que terminan realizando procesos de gestión de cartera para los deudores morosos y en algunos casos procesos jurídicos para recuperar el capital y los intereses causados por la operación de crédito.

1.1 Planteamiento del problema

Actualmente, se han usado técnicas de aprendizaje de máquina sobre conjuntos de datos generados a partir de entidades financieras. En (Turkson et al., 2016) se analizan datos de un crédito bancario real utilizando algoritmos de clasificación supervisada y no supervisada con el fin de elegir cuáles de estos son los más adecuados para predecir la solvencia crediticia de un cliente. En (Aboobyda & Tarig, 2016) se presenta un modelo para clasificar el riesgo de crédito en el sector bancario con el fin de predecir el estado de los préstamos utilizando tres algoritmos de clasificación supervisada. Además, en (Rogério et al, 2016) se genera un modelo predictivo utilizando algoritmos de clasificación para ayudar a identificar a los clientes con mayor potencial de volver de una situación de morosidad a una situación normal.

UnoSoftnet S.A.S es una empresa de desarrollo de software que cuenta con más de siete entidades financieras como clientes a nivel nacional y que brinda una plataforma de negociación integral para el registro, liquidación y control de operaciones factoring y crédito. Actualmente, existe la necesidad por parte de las entidades financieras que son clientes de UnoSoftnet S.A.S de tener un mecanismo que permita predecir qué clientes pueden incurrir en morosidad. El problema particular a trabajar en esta tesis es la pérdida de dinero por parte de las entidades financieras causada por la insolvencia crediticia de sus clientes que conlleva a que ellos dejen de pagar las cuotas pactadas en la operación de crédito causando pérdidas financieras. Este problema se evidencia en un reporte dado por la Superintendencia Financiera según el cual en agosto de 2016 el valor de las deudas en mora con la banca sumó 13,2 billones de pesos. Además, se conoce que este saldo crece a un ritmo mayor que el de la colocación de nuevos créditos, en particular, los créditos vencidos crecieron siete veces más que los préstamos nuevos (Carlos Arturo García, 2016). Para esto, se propone

obtener un modelo de predicción, usando técnicas de aprendizaje supervisado, que permita conocer qué clientes podrían incurrir en morosidad y así aplicar estrategias persuasivas de pago para disminuir el riesgo de crédito.

La pregunta de investigación que orienta este trabajo de grado es: ¿es posible construir un modelo basado en algoritmos de clasificación supervisados que permita hacer un análisis predictivo sobre los clientes que puedan incurrir en mora a partir de datos suministrados por entidades financieras?

1.2 Justificación del problema

1.2.1 Justificación Académica

Académicamente el desarrollo de este trabajo de grado permitió aplicar los conocimientos adquiridos por las asignaturas vistas en la carrera desde la línea de fundamentos de programación hasta la línea de desarrollo de software, conceptos y técnicas en bases de datos que permite realizar el pre-procesamiento de los datos que se utilizan en este trabajo de grado. Así como la Inteligencia Artificial que permite aplicar técnicas y estrategias para construir un modelo predictivo para identificar clientes en la base de datos que puedan incurrir en un estado de mora.

1.2.2 Justificación Económica

El análisis del riesgo crediticio ha sido un enfoque fundamental en la industria financiera y poder contar con un módulo de software que permita predecir posibles deudores morosos es importante para la evaluación de las carteras en términos de riesgo crediticio. Esto permite garantizar la sostenibilidad de la entidad financiera aplicando métodos y/o modelos que ayuden a minimizar estos riesgos.

1.3 Objetivos

1.3.1 Objetivos Generales

Desarrollar un prototipo para la gestión de riesgo crediticio utilizando técnicas de inteligencia artificial.

1.3.2 Objetivos Específicos

1. Construir una bodega de datos con la información de las transacciones y clientes de entidades financieras.
2. Aplicar técnicas de clasificación supervisada para construir modelos predictivos que permitan identificar clientes que puedan pasar a un estado de morosidad.

3. Implementar un prototipo para la gestión de riesgo crediticio.

4. Realizar las pruebas del prototipo.

1.4 Resultados Obtenidos

En la Tabla 1 se muestran los resultados obtenidos por cada objetivo específico.

Objetivo Específico	Producto(s) Obtenido(s)
Construir una bodega de datos sobre la información contenida en la base de datos de las entidades financieras.	Data mart generado de un modelo estrella con tres dimensiones y la tabla de hecho. Sección 4.1 Selección de datos.
Aplicar diferentes técnicas de clasificación supervisada para construir modelos predictivos que permitan identificar clientes que puedan pasar a un estado de morosidad.	Tres modelos de predicción utilizando técnicas de aprendizaje supervisado (redes neuronales, árboles de decisión y soportes de máquinas vectoriales). Capítulo 4 sección 4.2 Cada modelo se evaluó a partir de validación cruzada con diez iteraciones y análisis ROC.
Implementar un prototipo del módulo de gestión de riesgo crediticio que permita predecir los clientes que pueden pasar a un estado de mora.	Prototipo del módulo de gestión de riesgo crediticio. Capítulo 6
Realizar las pruebas del prototipo.	Reporte de pruebas unitarias del módulo de gestión de cartera. Capítulo 7

1.5 Alcances de la propuesta

Este trabajo de grado está dirigido para entidades financieras que trabajen con operaciones de crédito y se utilizan los datos de las empresas que son clientes de UnoSoftnet S.A.S. El prototipo no brinda información de qué estrategias tomar ante una cartera que pueda entrar en un estado de mora. Sobre el conjunto de datos se usan al menos tres técnicas de aprendizaje supervisado para la construcción de los modelos predictivos.

CAPÍTULO 2

MARCO TEÓRICO

En este capítulo se presentan los conceptos teóricos que sustentan este trabajo de grado orientado a técnicas de aprendizaje de máquina y desarrollo de software utilizando metodologías ágiles.

2.1 Aprendizaje de máquina

El aprendizaje de máquina es un campo basado en ideas de un conjunto diverso de disciplinas como la inteligencia artificial, la probabilidad y la estadística, la complejidad computacional, la teoría de la información, la psicología y la neurobiología, la teoría del control y la filosofía. En (Tom Mitchell, 1997) se da una definición muy útil y clara acerca del aprendizaje de maquina; "Se dice que un programa de computadora aprende de la experiencia E con respecto a alguna clase de tareas T y la medida de rendimiento P, si su desempeño en tareas en T, medido por P, mejora con la experiencia E". Uno de los objetivos del aprendizaje de máquina es poder construir sistemas capaces de adquirir el conocimiento necesario para realizar tareas, mejorando su rendimiento a través de la experiencia acumulada.

2.1.1 Minería de datos

La accesibilidad a grandes volúmenes de información y el uso de herramientas informáticas ha cambiado el análisis de datos orientándolo hacia determinadas técnicas especializadas que permiten descubrir nuevas y significativas relaciones, patrones y tendencias en los datos. Todas estas técnicas se engloban bajo el nombre de minería de datos. Las técnicas usadas en la minería de datos buscan el descubrimiento automático del conocimiento que se encuentra almacenado de forma ordenada en las grandes bases de datos. El objetivo de estas técnicas es descubrir patrones, perfiles y tendencias analizando los datos bajo tecnologías de reconocimiento de patrones, redes neuronales, algoritmos genéticos y otras técnicas avanzadas para el análisis de datos. Esto permite describir y comprender mejor la información almacenada y predecir comportamientos futuros (Pérez López, 2008).

2.1.2 Aprendizaje de máquina supervisado

Según (McCrea, 2014) el aprendizaje supervisado es una técnica del aprendizaje de máquina cuyos algoritmos utilizan un conjunto de datos de entrenamiento para hacer predicciones. El conjunto de datos de entrenamiento tiene datos de entrada y valores de respuesta con los cuales a partir de un algoritmo de aprendizaje supervisado se busca construir un modelo que permita hacer predicciones de los valores de respuesta para un nuevo conjunto de datos, teniendo la idea de que existe una relación entre los datos de entrada y el valor de respuesta.

Los algoritmos de aprendizaje supervisado buscan crear una función de predicción la cual es llamada hipótesis h(x). El aprendizaje consiste en usar algoritmos basados en procesos

matemáticos para optimizar la función h(x) que al pasarle como entrada el conjunto de datos x logre hacer una predicción precisa del valor de respuesta h(x).

Los problemas de aprendizaje supervisado se clasifican en problemas de regresión y clasificación. En un problema de regresión se trata de predecir resultados dentro de un resultado continuo, lo que denota asignar variables de entrada a alguna función continua. En un problema de clasificación se trata de predecir los resultados en un resultado discreto que en otras palabras se busca asignar variables de entrada en categorías discretas.

2.1.3 Redes neuronales artificiales

Según (Matich, 2001) la red neuronal artificial es un método de aprendizaje que proporciona un enfoque robusto para aproximar funciones objetivo de valor real, valor discreto y valor vectorial de cierto tipo de problemas donde se busca aprender a interpretar y procesar datos de comportamiento complejo. El estudio de redes neuronales artificiales está inspirado en el modelo de procesamiento de información en sistemas nerviosos biológicos que está formado por redes muy complejas de neuronas interconectadas. Especialmente, por la forma de procesamiento del cerebro humano que es totalmente distinta al procesamiento de un computador digital convencional ya que el cerebro humano es un sistema altamente complejo, no lineal y paralelo a diferencia de los computadores que son de tipo secuencial.

Un modelo de red neuronal no busca modelar exactamente el comportamiento fisiológico de una red biológica, sino más bien modelar las características relevantes en la interacción con toda la red. En la Figura 1 se puede ver los elementos básicos que componen una red neuronal.

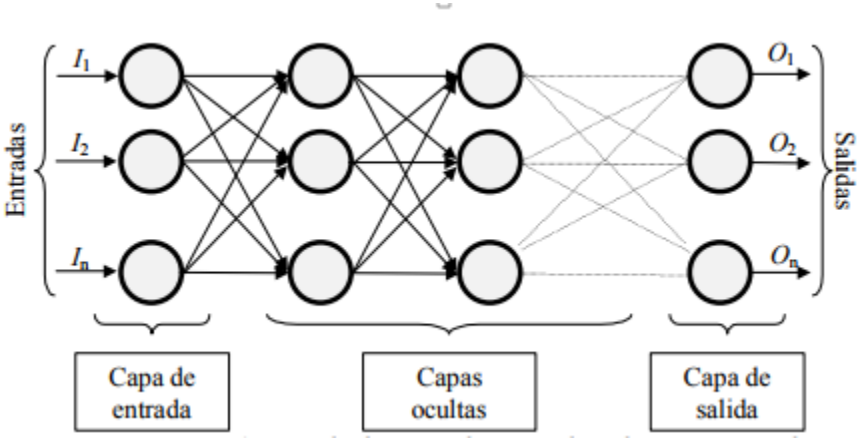


Figura 1. Elementos básicos de una red neuronal. Tomado de (Matich, 2001).

La red neuronal de la Figura 4 está constituida por neuronas totalmente interconectada divididas en tres capas (el número de capas puede variar). Los datos ingresan por la capa de entrada, luego pasan a través de la capa oculta y por último salen por la capa de salida. Las neuronas tratan muchos valores de entrada como si fueran un solo valor; a esto se le llama entrada global. La función de entrada permite combinar las entradas simples dentro de la entrada global y se calcula a partir del vector de entrada. Luego los valores de entrada se multiplican por los pesos anteriormente ingresados a la neurona. Las funciones de entrada comúnmente utilizadas son:

1. Sumatoria de las entradas pesadas donde se suma todos los valores de entrada con sus respectivos pesos.
2. Productoría de las entradas pesadas que se calcula como el producto de todos los valores de entrada a la neurona multiplicados por sus correspondientes pesos.
3. Máximo de las entradas pesadas donde se toma en cuenta solo el valor de entrada más fuerte previamente multiplicado por su peso.

Las neuronas artificiales también tienen estados de activación. La función de activación calcula el estado de activación de la neurona transformando la entrada global en un valor de activación donde normalmente el rango va desde 0 a 1 o de -1 a 1 donde 0 y -1 implica que la neurona está totalmente inactiva y 1 la neurona está activa. Algunas funciones típicas de activación son la función escalón, la función sigmoidea y la función tangente hiperbólica.

Por último, la neurona necesita una función de salida donde el valor proveniente de la función es el valor de salida de la neurona. Por lo tanto, este valor es el que se transfiere a las neuronas vinculadas. Si la función de activación está por debajo del umbral que activa la neurona ésta no transfiere ningún valor de salida. Los valores de salida deben estar en el rango [0,1] o [-1,1] y las dos funciones de salidas más utilizadas son la función identidad donde el valor de salida es el mismo de la entrada y la función binaria donde el valor de salida es 1 si el valor de activación es mayor o igual al umbral y 0 en caso contrario.

2.1.3.1 Algoritmo de retropropagación

Según (Matich, 2001) el algoritmo de retropropagación es el método de entrenamiento más utilizado en las redes neuronales y está compuesto por dos fases: primero se ingresa un patrón de datos de entrada, el cual se propaga por las diferentes capas de la red neuronal hasta producir una salida. La salida se compara con la salida deseada y se calcula el error cometido por las neuronas de salida. En segundo lugar, este error se tramite desde la capa de salida hacia la capa oculta donde cada neurona recibe un error que es proporcional a su contribución sobre el error total y de acuerdo a este error recibido se ajustan los pesos sinápticos de cada neurona.

2.1.4 Árboles de decisión

Según (Mitchell, 1997) uno de los métodos más utilizados y prácticos para la inferencia inductiva son los árboles de decisión. Este método aproxima funciones de valores discretos, es robusto para datos ruidosos y permite aprender expresiones disyuntivas. Los árboles de decisión clasifican instancias desde la raíz hasta el nodo hoja. Cada nodo en el árbol especifica una prueba de algún atributo de la instancia, y cada rama que desciende de ese nodo corresponde a uno de los valores posibles para este atributo. La instancia es clasificada iniciando desde el nodo raíz verificando el atributo especificado por este nodo, y luego bajando por la rama del árbol correspondiente al valor del atributo en el ejemplo dado. Estos pasos se repiten hasta llegar al nodo hoja.

Una gran cantidad de algoritmos que han sido desarrollados para construir árboles de decisión se basan en un algoritmo central que realiza una búsqueda codiciosa desde arriba hacia abajo a través

del espacio de posibles árboles de decisión. Este enfoque se presenta en el algoritmo ID3. El algoritmo básico ID3 construye el árbol de decisión de arriba hacia abajo, iniciando con el atributo que mejor clasifique los ejemplos de entrenamiento. Para esto es necesario evaluar cada atributo de la instancia a partir de una prueba estadística que determina qué tan bueno es este atributo para clasificar los ejemplos de entrenamiento. El atributo que mejor clasifique los ejemplos se selecciona y se usa como prueba para nodo raíz del árbol. Luego se crea un descendiente del nodo raíz para cada valor que pueda tomar este atributo. El valor del atributo se calcula a partir de la ganancia de información, este mide qué tanto un atributo determinado separa los ejemplos de entrenamiento de acuerdo con su función objetivo. El algoritmo ID3 usa esta medida para seleccionar un atributo y seguir expandiendo el árbol.

Para calcular la ganancia de información es necesario primero definir una medida comúnmente utilizada en la teoría de la información llamada entropía. La entropía es la medida de incertidumbre que hay en un sistema. Es decir, la probabilidad de que suceda cada uno de los posibles resultados dada una determinada situación. La función de entropía más utilizada es la binaria y su expresión se muestra en la Ecuación 1.

$$\text{Entropía}(S) \cong -P\Phi \log_2 P\Phi - P\Theta \log_2 P\Theta \qquad \text{Ec. (1)}$$

donde $P\Phi$ es la probabilidad de que un evento suceda y $P\Theta$ es la probabilidad de que el otro evento suceda. Cuando todos los posibles resultados son equiprobables la entropía toma valor de 1 bit de información promedio, cuando solo hay un resultado posible la entropía toma un valor de 0 bits de información promedio y cualquier otra probabilidad toma valores entre 0 y 1 bit de información promedio.

La ganancia de información es la diferencia entre la entropía de un nodo y la entropía de uno de sus descendientes.

2.1.5 Máquinas de soporte vectorial

Las máquinas de soporte vectorial son un método basado en aprendizaje para resolver problemas de clasificación y regresión. Según (Vapnik, 1998) las máquinas de soporte vectorial implementan la idea de mapear vectores de entrada X en el espacio Z de características de alta dimensión por medio de un mapeo no lineal, elegido a priori. En este espacio se construye un hiperplano de separación óptimo o un conjunto de hiperplanos para generar una separación entre las clases que permitan una clasificación correcta.

2.2 Metodología de desarrollo ágil

En febrero de 2001, nace el término "ágil" aplicado al desarrollo de software tras una reunión celebrada en Utah-EEUU. En esta reunión se logró esbozar valores y principios que permitieran a los equipos desarrollar software rápidamente respondiendo a los cambios que puedan surgir a lo largo del proyecto. Lo que se pretende con este tipo de métodos es brindar una alternativa a los procesos de desarrollo de software tradicionales que se caracterizan por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas.

En la reunión se generó un documento que resume la filosofía de la metodología ágil donde el individuo y las interacciones del equipo de desarrollo están por encima del proceso y las herramientas. La documentación debe ser corta y estar centrada en lo fundamental para la toma de decisiones. Es importante que se genere una colaboración entre ambas partes, el cliente y el equipo de desarrollo para asegurar el éxito del proyecto. Por último, es importante responder a los cambios que puedan surgir en el proyecto que permita determinar el éxito del mismo (Canós et al, 2003).

2.2.1 Programación extrema

Según (Beck k, 1999) la programación extrema es una metodología ágil que está centrada en potenciar las relaciones interpersonales como clave para el éxito en un desarrollo de software, se promueve el trabajo en equipo y se enfoca en el aprendizaje de los desarrolladores propiciando un buen clima de trabajo. La retroalimentación continua entre el cliente y el equipo de desarrollo, la simplicidad en las soluciones implementadas, la comunicación fluida entre el equipo de desarrollo y la habilidad para enfrentar los cambios en el proyecto son parte importante en la metodología.

Las características esenciales de la programación extrema están definidas en tres apartados principales: historias de usuarios, roles, procesos y prácticas. Las historias de usuario permiten especificar los requisitos del software donde el cliente describe brevemente las características del sistema ya sean requisitos funcionales o no funcionales. La propuesta de los roles en la programación extrema es: programadores, cliente, encargado de pruebas, encargado se seguimiento, entrenador, consultor y gestor. El ciclo de desarrollo de la programación extrema se define en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1

En la programación extrema se busca disminuir la mítica curva exponencial del costo de cambio a lo largo del proyecto. Esto se consigue gracias a las tecnologías disponibles para el desarrollo de software y las prácticas que se aplican a la metodología como el juego de planificación, las entregas pequeñas, la metáfora que describa cómo debería funcionar el sistema, el diseño simple, las pruebas, la refactorización que remueve la duplicación de código y mejora su legibilidad, la

programación en parejas, la propiedad colectiva del código que permite realizar los cambios en cualquier momento, la integración continua, el cliente in-situ que brinda la comunicación en todo momento del desarrollo y los estándares de programación que enfatiza la comunicación entre los programadores a través del código.

CAPÍTULO 3

ESTADO DEL ARTE

En este capítulo se muestran algunos antecedentes encontrados que trabajan el problema de la solvencia crediticia en entidades financieras y bancarias utilizando técnicas de inteligencia artificial.

3.1 Un enfoque de aprendizaje de máquina para predecir solvencia crediticia bancaria

El trabajo presentado en (Turkson et al., 2016) tiene como propósito predecir la solvencia crediticia a partir de una base de datos bancaria empleando diversos enfoques de aprendizaje de máquina. Además, se realiza un análisis comparativo para comprender los algoritmos que tuvieron mejor desempeño en la predicción. Como objetivo se busca identificar las características más importantes que determinan la solvencia crediticia, y desarrollar un modelo predictivo utilizando un enfoque de regresión lineal ordinaria. Estos enfoques permiten tener una mejor perspectiva sobre cómo hacer un análisis integral de la base de datos bancaria.

El conjunto de datos que se utiliza en este trabajo se obtuvo del repositorio de aprendizaje de máquina UCI (I-Cheng Yeh, 2016) y contiene 23 variables explicativas y una variable binaria de respuesta que describe si el cliente incumple, o no, con el pago (Si = 1, No = 0). Las variables son descritas en la Tabla 2.

Tabla 2. Describe los atributos del conjunto de datos. Adaptado de (I-Cheng Yeh, 2016).

Nº	**Atributo**	**Descripción**	**Tipo de dato**
1	X1	Monto del crédito otorgado en dólares	Numérico
2	X2	Género (Masculino = 1, Femenino = 2).	Nominal
3	X3	Educación (Escuela de postgrado = 1, Universidad = 2, Escuela secundaria = 3, Otro = 4).	Nominal
4	X4	Estado Civil (Casado = 1, Soltero = 2, Otro = 3)	Nominal
	X5	Edad (Años)	Numérico
	X6 a X11	Historial de los últimos 6 meses de pago (Abril a Septiembre del 2005) clasificados por estado de amortización (pago cumplido de la cuota = 1, retraso de pago por un mes = 1, retraso de pago por dos meses = 2,... retraso de pago por ocho meses = 8, retraso de pago por nueve meses o más = 9)	Nominal
	X12 a X17	Importe de estado de cuenta en dólares de abril a septiembre del 2005.	Numérico
	X18 a X23	Monto pagado por mes de abril a septiembre del 2005.	Numérico

El desafío que enfrentan las entidades bancarias y financieras con la insolvencia crediticia genera un riesgo económico ya que el cliente deja de cumplir con sus obligaciones pactadas. Sin embargo, una de las principales razones de los establecimientos bancarios es promover préstamos a los clientes, pero establecer quién es digno del crédito sigue siendo un desafío continuo en el sector bancario. Otro problema que enfrentan las entidades bancarias es el fraude crediticio, el cual puede ser tratado con los enfoques del aprendizaje de máquina ya que permite identificar patrones en las transacciones bancarias para distinguir entre una actividad fraudulenta y una que no lo es.

Sobre los datos se aplican varias técnicas de exploración para comprender la naturaleza de esta. El análisis exploratorio revela que hay una relación entre la edad del cliente, el monto del crédito otorgado, y su capacidad para pagar la cuota del siguiente mes donde los clientes entre veinte y sesenta años con el menor monto son los que más incumplen con el pago de sus préstamos. Esta relación se muestra en la Figura 2 donde el eje x representa el monto de crédito otorgado en dólares y el eje y representa la edad de los clientes. Los puntos azules representan los clientes que incumplen con el pago del próximo mes y los puntos rojos el caso contrario.

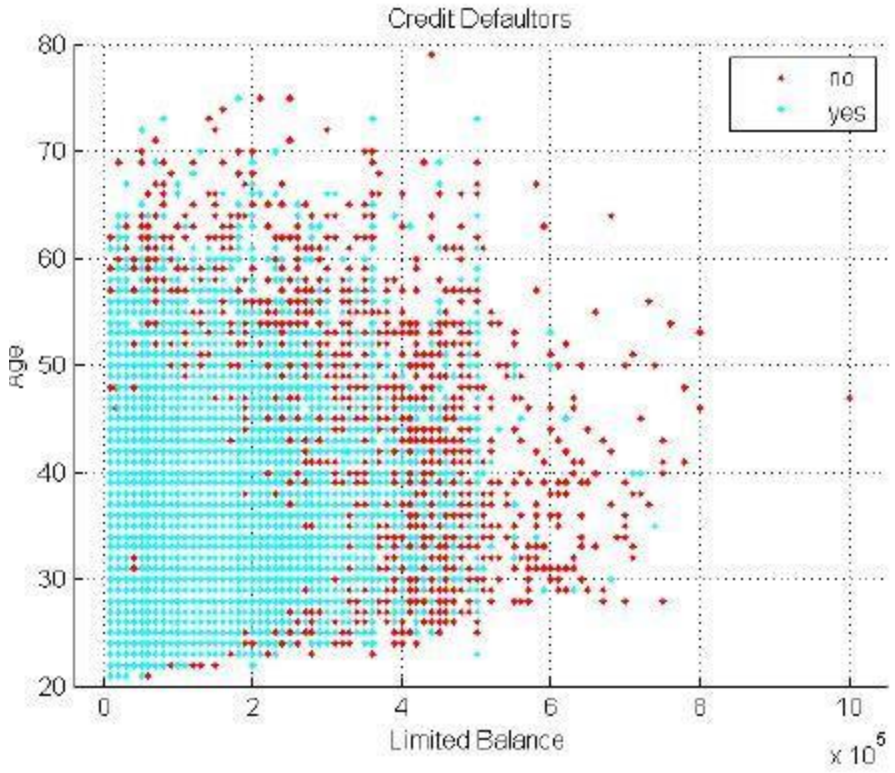


Figura 2. Relación entre la edad del cliente, el capital financiado y el incumplimiento del pago crediticio. Tomado de (Turkson et al., 2016).

Al aplicar diferentes algoritmos de aprendizaje se determinó cuáles son los más adecuados para estudiar el conjunto de datos, revelando que además del centroide más cercano y el Gaussian Naive Bayes el resto de algoritmos funcionan de manera apropiada en términos de precisión y otras métricas de evaluación obteniendo una tasa de predicción entre el 76% y más del 80%. En la Figura 2 se relacionan los algoritmos aplicados con sus correspondientes porcentajes.

También se determinó las características más importantes que influyen en la solvencia crediticia de los clientes. En la Figura 3 se relacionan las características del conjunto de datos con su nivel de importancia. Se observa que las cinco características más importantes son suficientes para predecir la solvencia crediticia. Para determinar esto se aplicaron algoritmos de aprendizaje de máquina a estas cinco características. Los resultados no muestran ninguna diferencia significativa entre el uso de las veintitrés características y las cinco características más importantes.

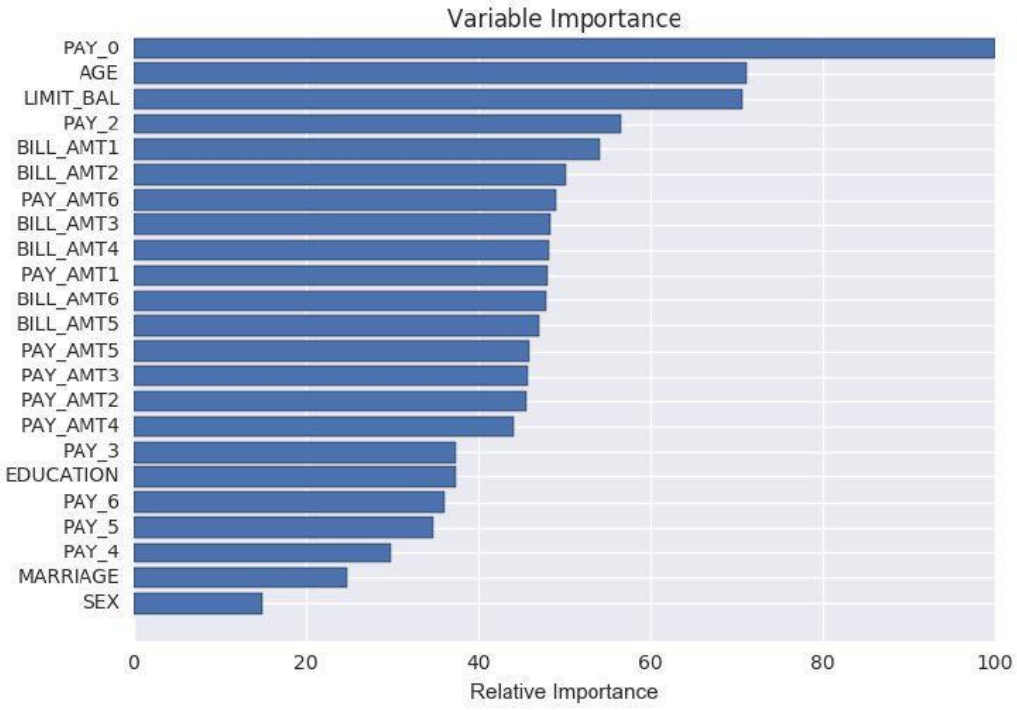


Figura 3. Relación de las características del conjunto de datos y su nivel de importancia. Tomado de (Turkson et al., 2016).

Utilizando las tres características más importantes que se muestran en la Figura 2 se desarrolla un modelo predictivo utilizando regresión lineal ordinaria. Este modelo se muestra en la Ecuación 2 donde y es la variable de respuesta que describe si el cliente incumple o no con el pago (Si = 1, No = 0).

$$y=0.4504+0.987PAY\_0+0.0018AGE-0.0227(\log (LIMIT\_BAL)) \quad \text{Ec. (2)}$$

Según los resultados obtenidos en esta investigación es necesario desarrollar un modelo híbrido de aprendizaje de máquina incorporando las características más importantes que permita mejorar la tasa de predicción y formular un sistema automatizado de riesgos bancarios.

3.2 Desarrollo de un modelo predictivo de riesgos en préstamos bancarios usando minería de datos.

El trabajo presentado en (Aboobyda & Tarig, 2016) tiene como objetivo desarrollar un modelo de predicción que permita clasificar el riesgo crediticio mediante técnicas de minería de datos. El modelo se construye usando datos del sector bancario y su propósito es predecir el estado de los préstamos como buenos o malos donde un préstamo bueno es aquel donde cliente tiene solvencia crediticia y malo donde carece de ésta.

El número de transacciones en el sector bancario crece rápidamente representando el comportamiento de los clientes y los riesgos entorno al préstamo. Existen muchos riesgos relacionados con los créditos bancarios, tanto para el banco como para el que obtiene el crédito. El riesgo en préstamos bancarios implica: riesgo crediticio, riesgo de liquidez y riesgo de tasas de interés. En este trabajo se enfocan en el riesgo crediticio que conlleva a una posible pérdida económica por parte del banco como consecuencia del incumplimiento de las obligaciones contractuales del cliente. En particular, se seleccionó una colección de datos del sector bancario que consta de ocho atributos los cuales son descritos en la Tabla 3.

Tabla 3. Describe los atributos de la colección de datos. Adaptado de (Aboobyda & Tarig, 2016).

Nº	Atributo	Descripción	Tipo de dato
1	Historial de crédito	Historial del anterior crédito del cliente	Nominal
2	Propósito	Propósito del préstamo	Nominal
3	Genero	Genero del cliente	Nominal
4	Monto de Crédito	Capital financiado (Monto del crédito)	Numérico
5	Años	Edad	Numérico
6	Alojamiento	Casa propia, arrendada, gratis	Nominal
7	Trabajo	El cliente actualmente tiene trabajo	Nominal
8	Clase	Clase de préstamo Bueno o Malo	Nominal

En la investigación se utilizaron tres algoritmos de clasificación para construir tres modelos diferentes. Estos algoritmos son: j48, bayesNet y naiveBayes. El conjunto de datos original se dividió en dos grupos, el conjunto de entrenamiento que representa el 80% de todos los datos y el conjunto de pruebas que representa el 20% del conjunto de datos. La exactitud de las técnicas J48, BayesNet y NaiveBayes son 78,3784%, 77,4775%, y 73,8739%, respectivamente.

Después de aplicar los tres algoritmos j48, bayesNet y naiveBayes, mediante el uso de la aplicación Weka se encontró que el mejor algoritmo para la clasificación de préstamos es el algoritmo j48 ya que tiene una alta precisión. El porcentaje de instancias correctamente clasificadas es de 78,3784%, las instancias incorrectamente clasificadas es 21,6216% y el error absoluto medio es de 0,3438.

3.3 Prediciendo el restablecimiento de operaciones crediticias en un banco Brasilero

En (Rogério et al, 2016) se presenta un estudio realizado en un banco brasileño que tiene como objetivo aplicar técnicas de minería de datos para construir un modelo predictivo. El propósito de este modelo predictivo es ayudar a los gerentes de cuentas de la institución bancaria a identificar los clientes con mayor potencial para pasar de una situación de morosidad a una situación normal entendiendo como situación de morosidad al incumplimiento de las obligaciones de pago por parte del cliente.

Desde enero de 2015 se ha presentado una caída en el suministro de crédito y un aumento en el incumplimiento de los pagos por parte de los clientes, esto como resultado de la disminución de las actividades económicas y la confianza de los inversores. Según el banco central de Brasil, los créditos a personas físicas son los que han mostrado el mayor crecimiento de incumplimiento pasando del 5,1% en diciembre de 2015 al 6,7% en abril de 2016.

En relación con este problema existen muchos estudios que clasifican al cliente debido al riesgo de crédito que representan separándolos en buenos y malos pagadores. Sin embargo, una vez que se presenta la situación de morosidad, se ha investigado poco para clasificar la posibilidad de que estos malos pagadores se conviertan en buenos pagadores de nuevo. Por lo tanto, el problema abordado en este artículo es el poder identificar los clientes que pueden pasar de una situación de morosidad a una situación normal de crédito. Para el artículo se utilizaron tres bases de datos. La primera base de datos contiene una muestra de 22.764 transacciones realizadas a finales de febrero de 2016 y contiene variables relacionadas con los datos de contratación como la fecha de operación, tiempo del modelo de operación, capital financiado, saldo pendiente y días de mora, totalizando 38 variables. La segunda base de datos contiene el estado de cada transacción y también el número de días con la operación vencida a corte de marzo de 2016. La tercera base de datos está compuesta por 158 variables con información demográfica y financiera de los clientes.

En este trabajo se desarrolló una actividad de preparación de datos donde las tres bases de datos se redujeron a solo una que contiene 196 variables. En el análisis inicial de los datos se encuentra un caso de clases desequilibradas ya que de las 22764 operaciones que tuvieron pagos atrasados solo 1548 regresaron a una situación regular. Se desarrollaron tres modelos predictivos para compararlos y elegir el mejor modelo que permitiera predecir qué clientes pueden salir del estado moroso. Inicialmente los modelos se procesaron en el software R. Sin embargo, por la gran cantidad de variables (196), el procesamiento de los datos tardaba demasiado y comprometía la eficiencia del estudio así que se utilizó el paquete H2O, integrada con R, para explorar el modo de red y los modelos de procesamiento paralelo. El procesamiento paralelo permitió que estos modelos se construyeran al mismo tiempo, lo cual aumentó eficiencia. Esto hizo posible construir más modelos con mejores parámetros de ajuste.

En este trabajo se usaron los siguientes algoritmos:

- GLM: Modelado lineal generalizado.
- GBM: Método de aumento de gradiente.
- DRF: Bosque aleatorio distribuido.

Los tres modelos generados tuvieron un alto resultado de evaluación de la curva ROC. Sin embargo, considerando que las clases están desequilibradas se hace necesario evaluar un segundo indicador como el PTCC que indica el porcentaje de resultados verdaderos que se clasificaron correctamente. Al aplicar los dos criterios de evaluación AUC (área bajo la curva) y PTCC (porcentaje verdaderos correctamente clasificados) a los modelos de clasificación se obtiene que el algoritmo GLM obtiene un valor AUC de 0.956881 y un PTCC de 63.63%. El algoritmo GBM obtiene un valor AUC de 0.983650 y un PTCC de 84.65%. El algoritmo DRF alcanza un valor AUC de 0.978096 y un PTCC de 61.60%. Con los resultados obtenidos queda claro que el algoritmo GBM logró el mejor rendimiento en todas las métricas utilizadas.

3.4 Redes neuronales para la evaluación de riesgos crediticios

En (Khashman, 2010) se describe un sistema de evaluación de riesgo crediticio a partir de un modelo supervisado de red neuronal basado en retropropagación. El objetivo es abordar los problemas que se pueden presentar al diseñar un modelo de red neuronal artificial con aplicación a la evaluación del riesgo crediticio. En este trabajo se construyen tres redes neuronales con diferente número de nodos ocultos y bajo diferentes esquemas de aprendizaje para decidir si se aprueba o se rechaza una solicitud de crédito. El propósito es proponer un sistema de evaluación de crédito eficiente, rápido y fácil de usar, basado en los resultados de la investigación.

La calificación y evaluación del crédito es una de las técnicas analíticas clave en la evaluación del riesgo crediticio, que ha sido un área de investigación activa en la gestión de riesgo financiero. Uno de los modelos de predicción más utilizados se desarrolla a partir de redes neuronales y aunque su aplicación es exitosa en la clasificación y evaluación crediticia se pueden presentar problemas que no permitan ofrecer un sólido juicio sobre si un solicitante debe o no recibir un crédito. En este artículo se abordan los problemas que se presentan al desarrollar un modelo predictivo a partir de redes neuronales tales como el uso de una alta proporción de conjuntos de datos de entrenamiento a validación, la normalización de los datos de entrada y el costo computacional que puede implicar el uso de redes neuronales en aplicaciones financieras.

Para la implementación del sistema de evaluación de crédito propuesto, se usó el conjunto de datos de créditos disponible públicamente en el repositorio de datos de UCI Machine Learning (Asunción & Newman, 2007). El conjunto de datos tiene 20 atributos nominales y numéricos y contiene 1000 instancias. En la Tabla 4 se describen los atributos del conjunto de datos.

Tabla 4. Describe los atributos del conjunto de dato. Adaptado de (Khashman, 2010).

Atributo	Descripción	Tipo de dato
1	Estado de cuenta	Nominal
2	Duración del crédito en meses	Numérico
3	Historia de crédito	Nominal
4	Propósito	Nominal
5	Cuenta de crédito	Numérico
6	Cuenta de ahorros	Nominal
7	Empleo actual desde	Nominal
8	Tasa de pago en porcentaje del ingreso disponible	Numérico
9	Estado personal y genero	Nominal
10	Otros deudores / garantes	Nominal
11	Residencia actual desde	Numérico
12	Propiedad	Nominal
13	Edad en años	Numérico
14	Otros planes de pago	Nominal
15	Alojamiento	Nominal
16	Número de créditos existentes en este banco	Numérico
17	Trabajo	Nominal
18	Número de personas a cargo	Numérico
19	Tiene o no teléfono	Nominal
20	Trabajador extranjero	Nominal

En el artículo se implementa una red neuronal supervisada que se basa en el algoritmo de aprendizaje de retropropagación debido a su simplicidad de implementación y a la disponibilidad de datos para capacitar y validar este aprendizaje supervisado. La Figura 4 muestra la topología general del modelo de red neuronal para la evaluación del crédito.

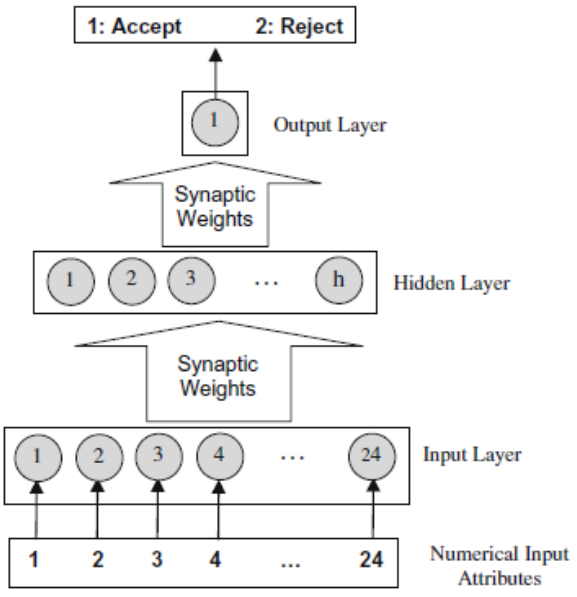


Figura 4. Topología general del modelo de red neuronal para evaluación crediticia. Tomado de (Khashman, 2010).

La capa de entrada de la red neuronal tiene 24 nodos que reciben valores numéricos normalizados. La capa oculta tiene h nodos; esta cantidad depende del modelo de red neuronal. En el artículo se implementaron tres modelos ANN-1 con h = 18, ANN-2 con h = 23 y ANN-3 con h = 27. La capa de salida tiene un solo nodo que representa un dato de salida binario; 0 para aceptar el crédito o 1 para rechazarlo. En la Tabla 5 se presentan los parámetros finales de los modelos de red neuronal.

Tabla 5. Describe los parámetros finales de los modelos de red neuronal artificial. Adaptado de (Khashman, 2010).

Modelo de red neuronal	ANN-1	ANN-2	ANN-3
Nodos en capa de entrada	24	24	24
Nodos en capa oculta	18	23	27
Nodos en capa de salida	1	1	1
Coeficiente de aprendizaje	0,0081	0,0095	0,0075
Momentum	0,70	0,69	0,79
Rango de peso inicial aleatorio	-0,3 a +0.3	-0,3 a +0.3	-0,3 a +0.3
Error mínimo requerido	0,008	0,008	0,008
Error obtenido	0,007972	0,008000	0,008531
Máximas iteraciones permitidas	25.000	25.000	25.000
Iteraciones realizadas	19.236	18.652	25.000
Relación óptima de entrenamiento-validación	500:500	400:600	600:400

Según (Khashman, 2010), el modelo de red neuronal ANN-2 funciona mejor cuando se utiliza 400 casos para entrenamiento y 600 casos para validación (40% y 60%) obteniendo una tasa de predicción total de 83,6%. El entrenamiento de este modelo se completó en aproximadamente 184 segundos, mientras que el tiempo en la toma de decisiones fue 5,17 × 10-5 segundos teniendo en cuenta las características de la máquina que cuenta con dos gigas de memoria RAM y un sistema operativo Win XP.

CAPÍTULO 4

MODELOS PROPUESTOS PARA LA PREDICCIÓN DE RIESGO CREDITICIO

En este capítulo se presentan los modelos propuestos para la predicción de riesgo crediticio y la selección de datos que se llevó a cabo para entrenar los modelos. Los modelos se generaron a partir de tres algoritmos de clasificación supervisada (redes neuronales, árboles de decisión y máquinas de soporte vectorial).

4.1 Selección de datos

Para la selección de los datos se cuenta con una base de datos relacional anónima la cual hace parte de la aplicación financiera que administra el ciclo de vida de los créditos registrados en una empresa cliente de UNOSOFT S.A.S. Inicialmente se procede a identificar la fuente de datos origen en la base de datos. Para el problema de gestión de riesgo crediticio es necesario obtener los datos básicos de los clientes con información de la operación crediticia y la información detallada de los pagos realizados en un periodo de tiempo. Para esto la base de datos cuenta con las tablas Clientes, Operaciones y DetIngresosClientes con las cuales se procede a realizar el diseño del modelo estrella en el entorno integrado SSMS (SQL Server Management Studio).

En el diseño del modelo estrella se identificó la tabla de hechos y las dimensiones. La tabla DetIngresosClientes es la tabla de hechos y contiene en detalle los pagos de las cuotas de cada cliente como el valor pagado de la cuota, los días de mora, el saldo capital de la cuota cancelada y fechas de cancelación. Las tablas dimensión están compuestas por la tabla clientes que contiene información básica de los clientes, la tabla operación que tiene información básica de las operaciones de crédito y la tabla tiempo que contiene fechas con un nivel de granularidad de días.

La Figura 5 muestra el modelo estrella generado a partir del análisis del proceso de pagos crediticios donde la tabla de ingresos corresponde a la tabla de hechos, DimOperaciones, DimClientes y DimFecha corresponden a las dimensiones del Data Mart que fue generado en el entorno integrado SSMS (Sql server management studio).

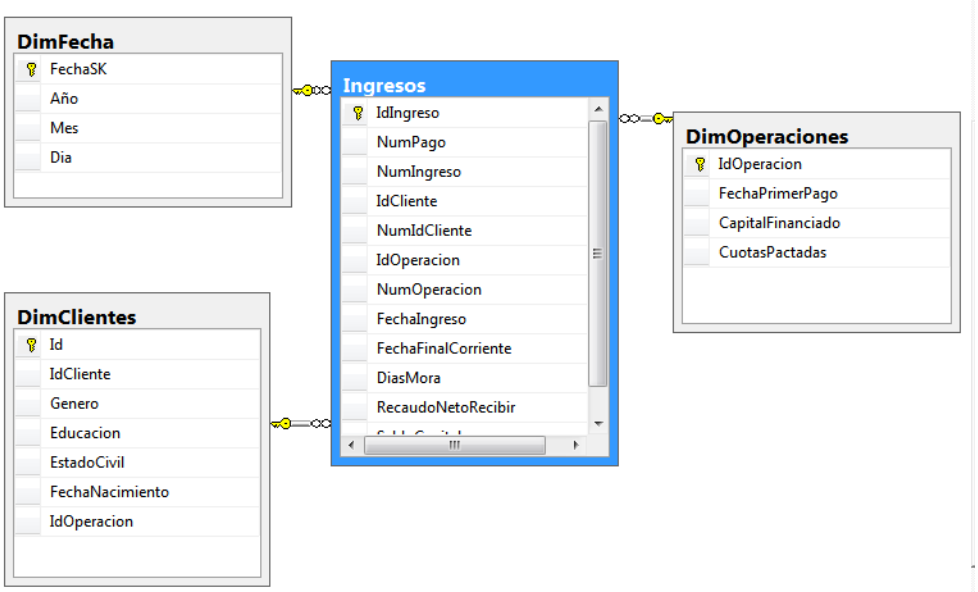


Figura 5. Modelo estrella para la selección de datos.

El proceso de transformación, extracción y carga de datos se realizó a través de procedimientos almacenados que extraen información válida de la base de datos y transforman datos cualitativos nominales categorizándolos numéricamente para la carga de datos al data mart.

La generación de los datos para entrenamiento y prueba de los modelos de aprendizaje supervisado se hizo por medio de un procedimiento almacenado con el cual se obtuvieron 561 registros y la información fue almacenada en un archivo CSV. Los 26 atributos que se usaron para generar los modelos pertenecen a datos básicos de los clientes, de las operaciones y de los pagos generados y se muestran en la Tabla 6. El atributo capital financiado indica el valor en pesos del crédito aprobado por la entidad financiera. Los estados de amortización de los últimos seis pagos registrados indican si cada pago se generó antes de vencimiento, en fecha de vencimiento o si se pagaron con mora. El importe de estado de cuenta indica el saldo en capital después de cada pago. La etiqueta de clase que se quiere predecir corresponde al pago después de los últimos seis meses y se obtiene de los ingresos del cliente categorizando el pago del mes número siete. Si hay un recaudo se asigna cero lo cual significa que el cliente no incumplió el pago de la cuota y si no hay un recaudo se asigna uno lo cual significa que el cliente incumplió el pago de la cuota.

Tabla 6. Atributos del conjunto de datos generados del data mart.

Nº	Atributo	Descripción	Tipo de dato
1	X1	Capital financiado en pesos colombianos	Numérico
2	X2	Género (Masculino = 1, Femenino = 2).	Nominal
3	X3	Educación (Primaria = 1, Secundaria = 2, Universitaria = 3, Maestría = 4).	Nominal
4	X4	Estado Civil (Casado = 1, Soltero = 2, Otro = 3)	Nominal
	X5	Edad (Años)	Numérico
	X6 a X11	Historial de los últimos 6 meses de pago clasificados por estado de amortización (pago antes de vencimiento = -2, no pago = -1, pago cumplido de la cuota = 0, retraso de pago por un mes = 1, retraso de pago por dos meses = 2,... retraso de pago por ocho meses = 8, retraso de pago por nueve meses = 9 etc.)	Nominal
	X12 a X17	Importe de estado de cuenta en pesos colombianos de los últimos 6 meses	Numérico
	X18 a X23	Monto pagado de los últimos 6 meses	Numérico
	X24	Etiqueta a predecir que corresponde al incumplimiento de la cuota siguiente a los últimos 6 meses de pago (cuota no incumplida = 0, cuota incumplida = 1).	Nominal

4.2 Construcción de los modelos

4.2.1 Modelo de predicción utilizando redes neuronales

Para la construcción del modelo usando redes neuronales se utilizó una red neuronal perceptrón multicapa (MLP) y algunas reglas adhoc para el diseño de la topología. Según (Flórez & Fernández, 2008) uno de los principales problemas de construir un modelo de red MLP es el diseño de la topología la cual es muy importante ya que permite la generalización del modelo. Según Lippman, (1987), las redes con una sola capa oculta resultan suficientes para resolver problemas arbitrariamente complejos siempre que el número de nodos ocultos sea al menos tres veces el número de nodos de entrada.Por otra parte (Hecht-Nielsen, 1990) aplica una extensión del teorema de Kolmogorov para demostrar que una red con una capa oculta compuesta por 2N+1 neuronas ocultas donde N es el número de neuronas de entrada y con funciones de transferencias continuas no lineales y crecientes resulta óptimo para computar cualquier función continua de N variables de entrada.

Para determinar el número de neuronas ocultas de la red MLP se utilizaron dos reglas adhoc que según (Flórez & Fernández, 2008) no son matemáticamente justificables pero han demostrado un buen comportamiento en diversas aplicaciones. La primera regla adhoc utilizada fue la regla de la pirámide geométrica que se basa en la suposición de que el número de neuronas de la capa oculta

debe ser inferior al número de neuronas en la capa de entrada y mayor que el número de neuronas en la capa de salida donde el número de neuronas en cada capa sigue una progresión geométrica, tal que para una red de una sola capa oculta el número de neuronas en ella debe ser un aproximado de $\sqrt{N} * M$, donde N es el número de neuronas en la capa de entrada y M el número de neuronas en la capa de salida. La segunda regla adhoc utilizada se conoce como Capa oculta – Capa de entrada y consiste en relacionar el número de neuronas de la capa oculta con el número de neuronas en la capa de entrada. Normalmente suele aplicarse la regla 2*1 de forma que el número de neuronas ocultas no puede ser superior al doble del número de variables de entrada.

Para la construcción de la red perceptrón multicapa se utilizó la librería neuralnet del lenguaje R variando por medio de ciclos el número de capas ocultas y el número de nodos siguiendo las dos estrategias adhoc mencionadas anteriormente. Igualmente se cambia el valor del *threshold* que especifica el umbral para las derivadas parciales de la función de error como criterios de parada teniendo en cuenta desde qué umbral comienza a disminuir la sensibilidad en la predicción de los datos de prueba. La variación se realiza desde un 30% de error en el entrenamiento hasta un 10% de error.

Los datos usados para la construcción tienen una proporción de 28% de casos positivos que indica el porcentaje de clientes que incumplieron el pago de la cuota y 72% de casos negativos que indica el porcentaje de clientes que no incumplieron el pago de la cuota. Esta proporción desequilibrada se da porque son menos los clientes que incumplen que los que cumplen con el pago de sus cuotas. Estos datos fueron divididos 90% para entrenamiento y 10% para prueba con las mismas proporciones 28% de casos positivos y 72% de casos negativos aproximadamente. En total se generaron 998 redes MLP a partir de las reglas adhoc mencionadas anteriormente donde se generaron redes con rangos de nodos ocultos desde la raíz cuadrada del número de nodos de entrada ($\sqrt{N} * M$) hasta el doble del número de nodos de entrada más uno (2N+1) con una y dos capas ocultas y modificando el hiperparámetro *threshold* en rangos de 30% de error hasta 10% de error.

Sobre cada red se calculan los parámetros de exactitud, sensibilidad, especificidad, precisión (ver sección 5.1) y se seleccionan las diez redes con mejor desempeño en la sensibilidad como parámetro más importante para este problema seguido por la exactitud y la precisión por cada *threshold*. De las 30 redes seleccionadas se hace una validación cruzada con k=10 para calcular la exactitud promedio y validar qué tan bien se ajustan los datos a las redes neuronales. El diseño topológico de la red con mayor número de aciertos se presenta en la Figura 6 y cuenta con 24 nodos en la capa de entrada que corresponden a los atributos representados en la Tabla 6, dos capas ocultas donde la primera capa oculta tiene 34 nodos, la segunda capa oculta 25 nodos y la capa de salida con un nodo que representa la salida de tipo binario donde 1 es cuando el cliente incumple con el pago de la cuota pactada y 0 cuando no incumple con el pago de la cuota pactada.

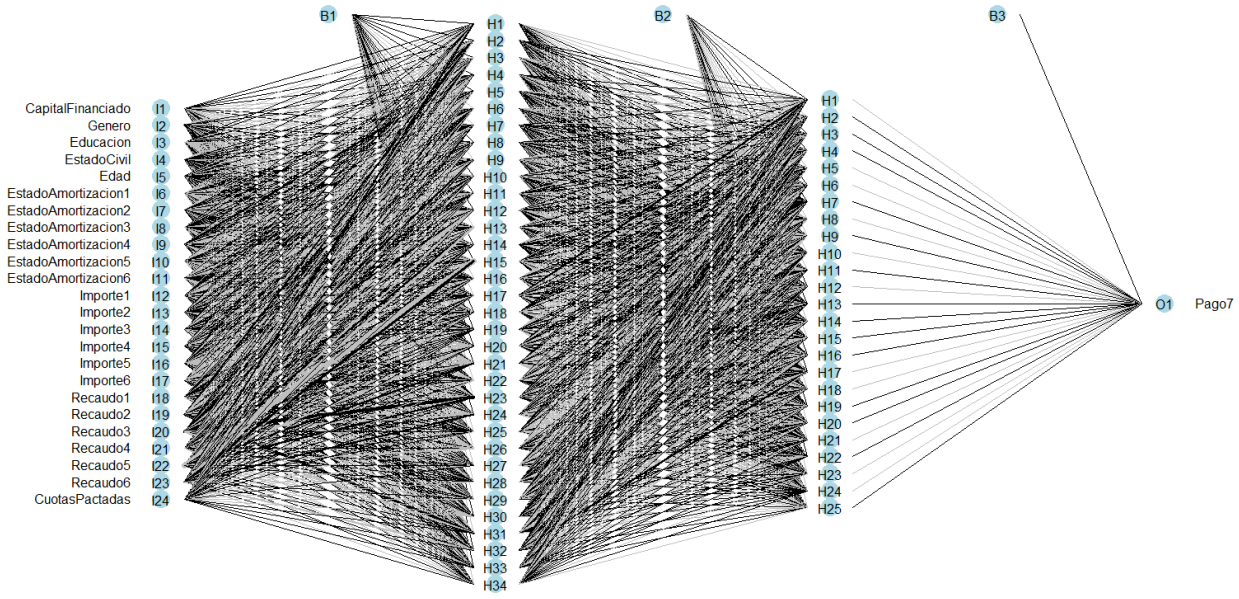


Figura 6. Modelo red MLP

4.2.2 Modelo de predicción utilizando árboles de decisión

En esta sección se presentan tres modelos de predicción de riesgo crediticio que se basan en tres algoritmos de árboles de decisión conocidos como C4.5, Random Forest y C5.0

4.2.2.1 Árboles de decisión aplicando el algoritmo C4.5

Para la construcción del modelo a partir de árboles de decisión aplicando el algoritmo C4.5 se utilizó la librería rpart del lenguaje R. Se experimentó con el hiperparámetro *minsplit* que es el número mínimo de observaciones que deben existir en un nodo para que se produzca o se intente generar una división. El valor por defecto de este hiperparámetro es 20 y se experimentó entre un rango del valor por defecto hasta el número máximo de observaciones en los datos de entrenamiento que son 504. El árbol de decisión generado y podado se muestra en la Figura 7 y consta de cinco pruebas sobre las variables independientes que están relacionadas en la Tabla 6 (EstadoAmortización6, Recaudo6, EstadoAmortización4). La clasificación de un nuevo cliente se realiza pasando los datos del cliente anteriormente mencionados por la prueba que hay en cada nodo siguiendo la rama del árbol que cumpla con las condiciones generadas en cada nodo hasta llegar al nodo hoja del árbol que representa la variable dependiente que se quiere predecir.

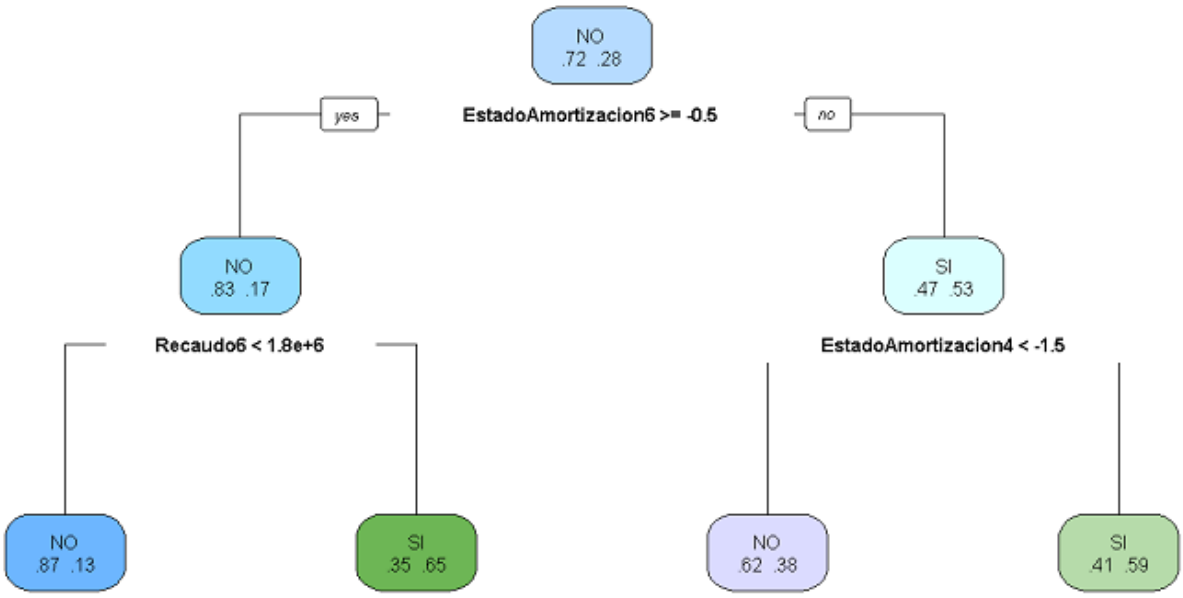


Figura 7. Modelo árbol C4.5

4.2.2.2 Árboles de decisión aplicando el algoritmo Random Forest

Para la construcción del modelo a partir de árboles de decisión aplicando el algoritmo random forest se utilizó la librería random forest del lenguaje R que tiene tres hiperparámetros que hay que definir, estos son ntree, mtry, y nodesize.

Para la generación del árbol de decisión se realiza una optimización de los hiperparámetros del algoritmo como el *ntree* que indica el número de árboles a generar. Para definir el *ntree* se realizó una validación cruzada probando diferentes valores iniciando por 1000 árboles hasta 2500 con el objetivo de evitar consumir recursos computacionales innecesarios. El valor más preciso para *ntree* fue 2500 con una precisión media de 75.21% y un coeficiente kappa de 31.48% que muestra una fuerza de concordancia mediana. La Tabla 7 muestra los resultados obtenidos de la validación cruzada para hallar el *ntree*.

Tabla 7. Exactitudes para diferentes valores del parámetro ntree

Ntree	*Precisión media*	*Coeficiente kappa*
1000	*0.7488*	*0.3079*
1500	*0.7494*	*0.3113*
2000	*0.7514*	*0.3158*
2500	*0.7521*	*0.3148*

El hiperparámetros *mtry* indica el número de variables tomadas de forma pseudoaleatoria para ser candidatos en cada división. Para esto se define una función que devuelve un arreglo de la tasa de error de los casos que no son considerados para entrenar el árbol al cual se le llama *out-of-bag-error* (error OOB). El hiperparámetro *mtry* va desde uno hasta la cantidad máxima de los predictores que son 24. Por último, se grafica para visualizar el menor error obtenido que se alcanzó c

on 11 variables.

La Figura 8 muestra la evolución del error OOB para diferentes valores de *mtry*. Se puede observar que en el 11 se obtiene el menor error.

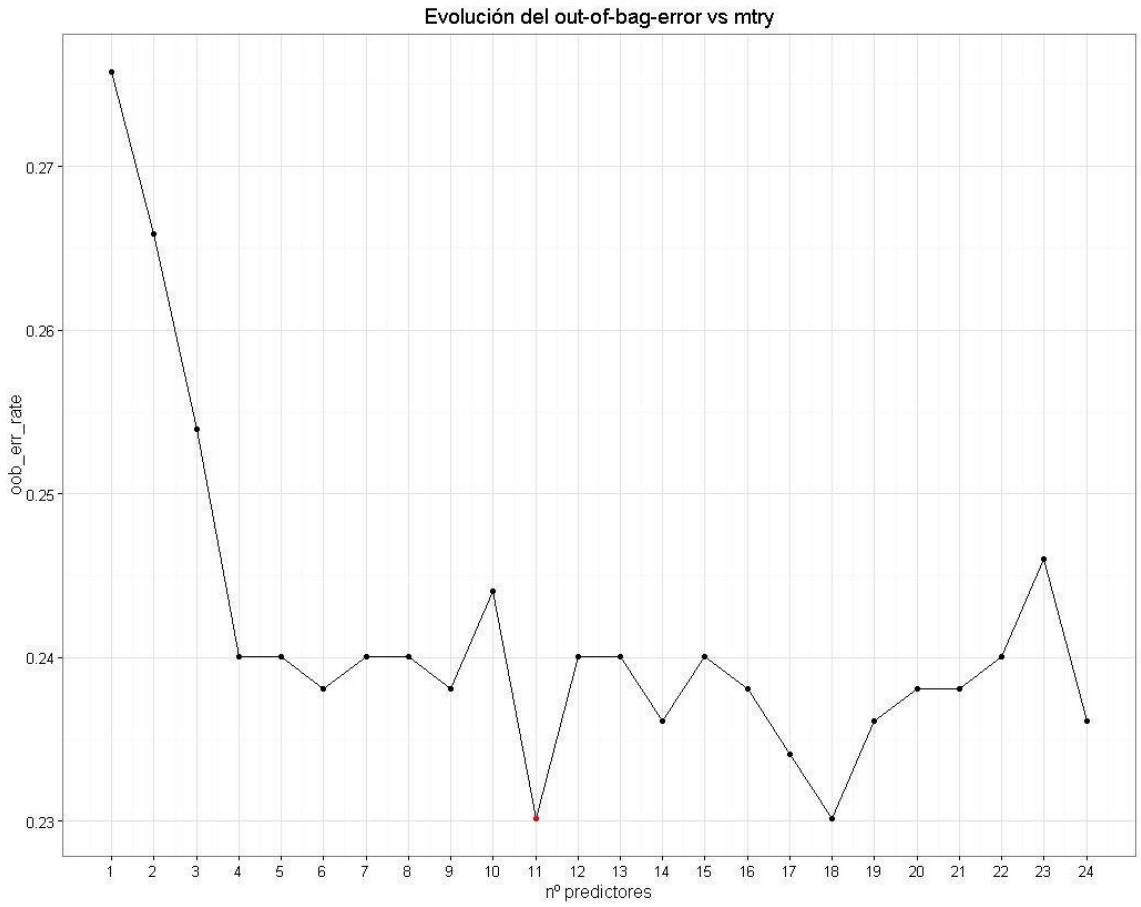


Figura 8. Evolución error OOB VS *mtry*

El algoritmo random forest tiene además el hiperparámetro *nodesize* que indica el tamaño mínimo de nodos terminales. De igual manera se define una función que genera un arreglo de error OOB por cada árbol generado en un rango que va desde 1, que es el valor por defecto para árboles de clasificación, hasta 20 donde se observa un incremento del error ya que al establecer un número grande en este parámetro se generan árboles más pequeños por ser un criterio de parada. La prueba se realiza con validación cruzada. La Figura 9 muestra la evolución del error OOB para diferentes valores de *nodesize*. El *nodesize* que muestra un menor error es 14.

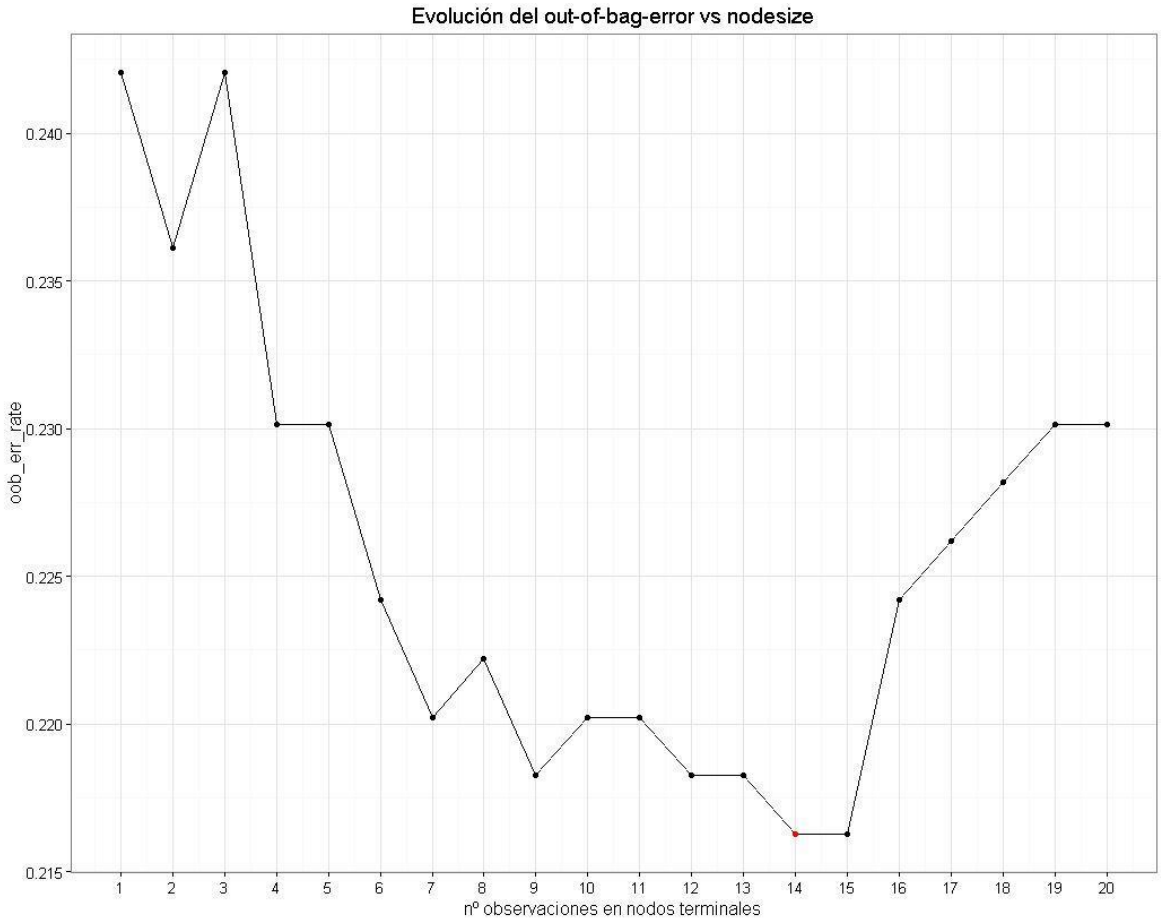


Figura 9. Evolución error OOB VS nodesize

4.2.2.3 Árboles de decisión aplicando el algoritmo C5.0

Para la construcción del modelo de predicción utilizando el algoritmo C5.0 se hace uso de la librería C50 en el lenguaje R. El algoritmo C5.0 tiene el hiperparámetro *costs* con el cual se puede

definir el peso de los errores para enfatizar ciertas clases sobre otras. Para el problema que se aborda en este trabajo de grado, realizar una predicción que clasifique al cliente como un cliente que no va a incumplir y termine incumpliendo es más costoso que un cliente que se clasifique como un cliente que incumplirá y que cumpla con el pago. Por tal motivo se define una matriz de costos con una dimensión de 2x2 ya que se cuenta con dos clases (clientes que pagan y clientes que no pagan). La matriz se presenta en la Tabla 8. La asignación del error más costoso se calcula a partir de una validación cruzada de 10 iteraciones variando el costo de 1 a 10 y generando la exactitud del modelo. El objetivo es encontrar un valor que permita mejorar la sensibilidad sin que decaiga demasiado la exactitud. Los resultados indican que los valores superiores a 4 en la clase clientes donde se predice que no incumplirá y terminan incumpliendo alcanzan exactitudes inferiores al 65%. A continuación se muestra la matriz de costos definida y los resultados obtenidos en la validación cruzada.

Tabla 8. Describe la matriz de costos definida como hiperparámetro para el modelo C5.0

	Real	
Predicho	NO	SI
NO	0	4
SI	1	0

Tabla 9. Resultados obtenidos por la validación cruzada de diez iteraciones por cada costo definido

costs	Promedio Exactitud
1	77.71%
2	72.48%
3	69.64%
4	65.88%
5	58.58%
6	55.95%
7	53.61%
8	49.04%
9	47.24%
10	45.14%

El árbol de decisión generado se muestra en la Figura 10 y consta de doce variables que están relacionadas en la Tabla 6 (EstadoAmortización6, Recaudo6, recaudo5, Edad, Educación, Recaudo4, EstadoAmortización5, EstadoAmortización4, Importe, EstadoAmortización2, Recaudo1, CuotasPactadas). La clasificación de un nuevo cliente se realiza pasando los datos del cliente anteriormente mencionados por las pruebas que hay en cada nodo siguiendo la rama del árbol que cumpla con las condiciones generadas en cada nodo hasta llegar al nodo hoja del árbol que representa la variable dependiente que se quiere.

Figura 10. Es el árbol generado utilizando el algoritmo C5.0

4.2.3 Modelo de predicción a partir de máquinas de soporte vectorial

Para la construcción del modelo de predicción a partir de máquinas de soporte vectorial se utilizó la librería e1071 del lenguaje R. Para encontrar la mejor configuración de los hiperparámetros se utiliza la función *tune* de la librería e1071 que permite realizar validación cruzada para ajustar los hiperparámetros de los métodos estadísticos utilizando una búsqueda de cuadrícula con lo cual se desea ajustar los hiperparámetros *cost* y *gamma* con diferentes kernel. El *cost* y el *gamma* son hiperparámetros que permiten variar el margen de separación entre observaciones que para este caso son 561 de los cuales 28% son de casos positivos (clientes que no pagan la cuota) y 72% de casos negativos (clientes que cancelan la cuota pactada). El *cost* es el peso que se le da a cada observación a la hora de clasificar que incide en el error cometido por la función de regresión, si este valor es muy grande mayor es el peso de una observación y el SVM sería más estricto en el momento de predecir ya que ajustaría perfectamente el hiperplano predictor al conjunto de datos de entrenamiento haciendo que el modelo se sobrestime. De lo contrario, un *cost* demasiado pequeño clasificaría erróneamente un número muy elevado de observaciones.

El gamma es otro hiperparámetro que permite suavizar la sobrestimación e influye en la distancia entre las observaciones que separan los subespacios del SVM. Un menor gamma implica mayor distancia entre las observaciones y por ende la estimación es más conservadora. Sin embargo, un valor de gamma muy elevado genera predicciones menos suavizadas y esto hace que el modelo se sobrestime. Para calcular estos hiperparámetros se utiliza la función tune a la cual se le proporciona los rangos del *cost* y el *gamma* que va desde 0,1 hasta 100 y los posibles kernel (lineal, radial, polinomial y sigmoid). El kernel con mejores resultados es el sigmoid que obtuvo 8 de *cost* y 4 de *gamma*.

CAPÍTULO 5
PRUEBAS Y ANÁLISIS DE RESULTADOS

En este capítulo se analizan los resultados de las pruebas realizadas a los modelos predictivos a partir de validación cruzada de 10 iteraciones con lo cual se busca validar qué tan bien se ajustan los datos a cada modelo. También se analizan los resultados de la experimentación con el tipo de prueba 90% y 10% y se procede a comparar los modelos a partir de un análisis ROC analizando el área bajo la curva de cada modelo.

5.1 Criterios de comparación

El criterio de comparación que se usó para los modelos predictivos de riesgo crediticio es el análisis de la curva ROC calculando el parámetro AUC (Área bajo la curva) ya que es un indicador de la capacidad predictiva de cada modelo. Los parámetros calculados en los resultados de cada modelo son los verdaderos positivos (VP) que son los casos en que la prueba clasifica un cliente que si incumple y realmente incumple, Falsos positivos (FP) que son los casos en que la prueba clasifica un cliente que si incumple y realmente no incumple, verdaderos negativos (VN) que son los casos en que la prueba clasifica un cliente que no incumple y realmente no incumple y falsos Negativos (FN) que son los casos en que la prueba clasifica un cliente que no incumple y realmente si incumple. Con estos parámetros se calcula la sensibilidad que es la probabilidad que el modelo clasifique correctamente un cliente que incumplirá el pago de su cuota, la especificidad que es la probabilidad que el modelo clasifique correctamente un cliente que no incumplirá el pago de su cuota, la exactitud que mide la fracción de predicciones correctas, la precisión mide la fracción de los verdaderos positivos entre los casos que se prevén positivos (VP+FP), la tasa de error que es el promedio de las clasificaciones incorrectas del modelo y el valor de predicción negativo (VNP) que es el porcentaje de las clasificaciones correctas de clientes que no incumplirán el pago de las cuotas. A continuación, se definen las medidas de exactitud que se utilizan en las pruebas.

Sensibilidad=VP/(VP+FN)

Especificidad=VN/(VN+FP)

Exactitud = VP+VN/(VP+VN+FN+FP)

Precisión = VP/(VP+FP)

Tasa de error = FP+FN/(VP+VN+FN+FP)

VNP = VN/(FN+FP)

La curva ROC representa 1-especificidad frente a la sensibilidad para cada punto de corte en la escala de resultados de los modelos. Cada resultado de predicción representa un punto en el espacio ROC y el mejor método posible de predicción se situaría en un punto en la esquina superior izquierda del espacio ROC. Un parámetro para evaluar la bondad de un modelo es el área bajo la curva ROC (AUC) ya que refleja qué tan bueno es el modelo para discriminar clientes que puedan

caer en un estado moratorio al no pagar la próxima cuota pactada. La curva ROC y el parámetro AUC se calcula en cada modelo con la librería ROCR del lenguaje R.

5.2 Pruebas y resultados en modelo de red neuronal

5.2.1 Validación cruzada con diez iteraciones, evaluación entrenamiento y prueba y análisis ROC

Para la evaluación de las redes neuronales generadas se utilizan dos tipos de pruebas: Validación cruzada con diez iteraciones para saber qué tan bien se ajustan los datos a los modelos calculando la exactitud promedio y la prueba 90% y 10% para calcular los criterios de evaluación descritos en la sección 5.1. De igual manera se realiza un análisis ROC al modelo seleccionado para saber la bondad del modelo y un análisis de sensibilidad para conocer el efecto o influencia de cada variable predictora sobre la variable dependiente[Fn1]. De las 998 redes neuronales MLP generadas se seleccionan las 10 redes con mejor rendimiento en sensibilidad seguido por exactitud por cada *threshold* (0.1, 0.2, 0.3). A las redes MLP se les realizó una validación cruzada con diez iteraciones sobre el conjunto de datos con el objetivo de obtener un promedio de la exactitud general y saber qué modelo se ajusta mejor a los datos. En la Tabla 10 se muestra los resultados obtenidos en la validación cruzada ordenado ascendentemente por exactitud.

Tabla 10. Validación cruzada 10 iteraciones modelos de red neuronal

Numero nodos 1ª Capa Oculta	Numero nodos 2ª Capa Oculta	*Threshold*	Exactitud promedio
7	0	0.1	68.29%
13	13	0.3	67.68%
29	0	0.3	66.69%
34	13	0.3	66.12%
22	9	0.3	66.10%
25	22	0.3	65.92%
33	8	0.1	63.64%
40	23	0.3	64.96%
13	0	0.1	64.88%
41	30	0.2	64.08%
34	16	0.3	63.78%
34	25	0.3	63.62%
39	20	0.3	63.58%
22	7	0.2	63.51%
29	26	0.2	63.51%
43	28	0.1	63.45%
25	8	0.1	63.37%
43	35	0.2	63.32%
30	21	0.2	63.13%
48	38	0.2	62.97%
38	23	0.2	62.92%
37	32	0.2	62.31%

29	22	0.3	61.90%
13	11	0.1	61.41%
20	6	0.1	61.39%
29	15	0.2	60.13%
42	37	0.1	59.29%
45	43	0.1	58.93%
25	20	0.1	58.23%
32	24	0.2	57.78%

En la Tabla 11 se muestra los resultados obtenidos de las 10 mejores redes MLP generadas por cada *threshold* (0.1, 0.2, 0.3) utilizando el tipo de prueba 90% y 10%. Los parámetros calculados son la exactitud, la tasa de error, la sensibilidad, la especificidad, la precisión y el promedio de verdaderos negativos (NPV). La tabla está ordenada de forma descendente por sensibilidad y exactitud donde en primer lugar está el modelo de red neuronal de la Figura 6 cuya exactitud promedio alcanzada es de 0.6362 que indica que el 63.62% de las predicciones realizadas en los datos de prueba son correctas y por tanto la tasa de error es de 36.38% que indica el porcentaje de clientes mal clasificados. La sensibilidad que es la métrica más importante en este problema alcanzó un 0.71833 que indica que el 71.83% de los clientes que incumplen el pago de sus cuotas serán clasificados correctamente. Este resultado es bueno considerando que los datos tienen una disparidad significativa entre el número de casos de clientes que incumplen los pagos de sus cuotas y clientes que no incumplen los pagos de sus cuotas. La precisión alcanzada por el modelo es de 0.4358, es decir, cuando el modelo predice un cliente que incumplirá el pago de su cuota, acierta el 43,58% de las veces. La especificidad alcanzada por el modelo es de 0.6834 que implica un 68.34% de probabilidad de que, dado un cliente que realmente no incumpla el pago de la cuota, el modelo llegará a la misma conclusión y el valor de predicción negativo alcanzó un 0.8715, es decir, que el 87.15% de los clientes que no incumplirán el pago fueron clasificados correctamente.

Tabla 11. Parámetros calculados a los modelos de red neuronal

N°Capa 1	N°Capa 2	Threshold	Sensibilidad	Especificidad	Exactitud	Tasa de error	Precisión	NPV
34	25	0.3	71.83	68.34	68.98	36,38	43.58	87.15
13	0	0.1	70.83	61.87	64.17	35,12	39.08	86
25	8	0.1	70.83	63.30	65.24	36,63	40	86.27
45	43	0.1	70.83	70.50	70.58	41,07	45.33	87.50
41	30	0.2	68.75	64.02	65.24	35,92	39.75	85.57
33	8	0.1	68.75	68.34	68.44	36,36	42.85	86.36
43	35	0.2	68.75	66.18	66.84	36,68	41.25	85.98
30	21	0.2	68.75	64.74	65.77	36,87	40.24	85.71
37	32	0.2	68.75	65.46	66.31	37,69	40.74	85.84
29	22	0.3	68.75	66.18	66.84	38,1	41.25	85.98
42	37	0.1	68.75	60.43	62.56	40,71	37.5	84.84
7	0	0.1	66.66	73.38	71.65	31,71	46.37	86.44
13	13	0.3	66.66	72.66	71.12	32,32	45.71	86.32
29	0	0.3	66.66	68.34	67.91	33,31	42.10	85.58
34	13	0.3	66.66	69.06	68.44	33,88	42.66	85.71

22	9	0.3	66.66	66.90	66.84	33,9	41.02	85.32
25	22	0.3	66.66	66.18	66.31	34,08	40.50	85.18
40	23	0.3	66.66	65.46	67.37	35,04	0.4	85.04
34	16	0,3	66.66	65.46	65.77	36,22	0.4	85.04
39	20	0.3	66.66	69.78	68.98	36,42	43.24	85.84
43	28	0.1	66.66	65.46	65.77	36,46	40.00	85.04
22	7	0.2	66.66	66.90	66.84	36,49	41.02	85.32
29	26	0.2	66.66	67.62	67.37	36,49	41.55	85.45
38	23	0.2	66.66	68.34	67.91	37,08	42.10	85.58
48	38	0.2	66.66	69.06	68.44	37,03	42.66	85.71
13	11	0.1	66.66	66.90	66.84	38,59	41.02	85.32
20	6	0.1	66.66	71.22	70.05	38,61	44.44	86.08
29	15	0.2	66.66	66.18	66.31	39,87	40.50	85.18
25	20	0.1	66.66	66.18	66.31	41,77	40.50	85.18
32	24	0.2	66.66	61.15	62.56	42,22	37.20	84.15

A continuación, se realiza un análisis ROC al modelo de red neuronal de la Figura 6. El análisis muestra la representación gráfica de la sensibilidad frente a la especificidad. La fracción de verdaderos positivos es la sensibilidad que es la probabilidad de clasificar correctamente un cliente cuyo estado real sea definido como positivo (si incumple con el pago de la próxima cuota) y la fracción de falsos positivos es la especificidad que es la probabilidad de clasificar correctamente a un cliente cuyo estado real sea clasificado como negativo (no incumple con el pago de la cuota). El parámetro área bajo la curva (AUC) se calcula para evaluar la bondad de la red neuronal, esta área posee un valor comprendido entre 0.5 y 1 donde el 1 representa un valor de predicción perfecto y 0.5 un modelo sin capacidad discriminatoria de predicción. El modelo de red neuronal de la Figura 6 alcanzó un 0.6991 de AUC que es considerado un resultado de test regular ya que existe un 69.91% de probabilidad de que la predicción realizada a un cliente que incumplirá el pago de la cuota sea más correcto que el de un cliente escogido al azar que no incumplirá el pago. La Figura 11 muestra la curva ROC del modelo de red neuronal con su parámetro AUC.

El análisis de sensibilidad se realiza utilizando la función Olden que utiliza el algoritmo de ponderaciones de conexión con la librería NeuralNetTools de R. El análisis arroja que las variables Recaudo3 y Género tienen las relaciones negativas y positivas más fuertes con respecto a la variable de respuesta que es si el cliente incumple el pago de su cuota. De igual manera las variables EstadoAmortizacion5, EstadoCivil y CuotasPactadas tienen una importancia relativa cercana a 0 que indica que tienen poca importancia para la variable dependiente. La Figura 12 muestra la importancia relativa de las variables independientes en el modelo de red neuronal seleccionado.[Fn2]

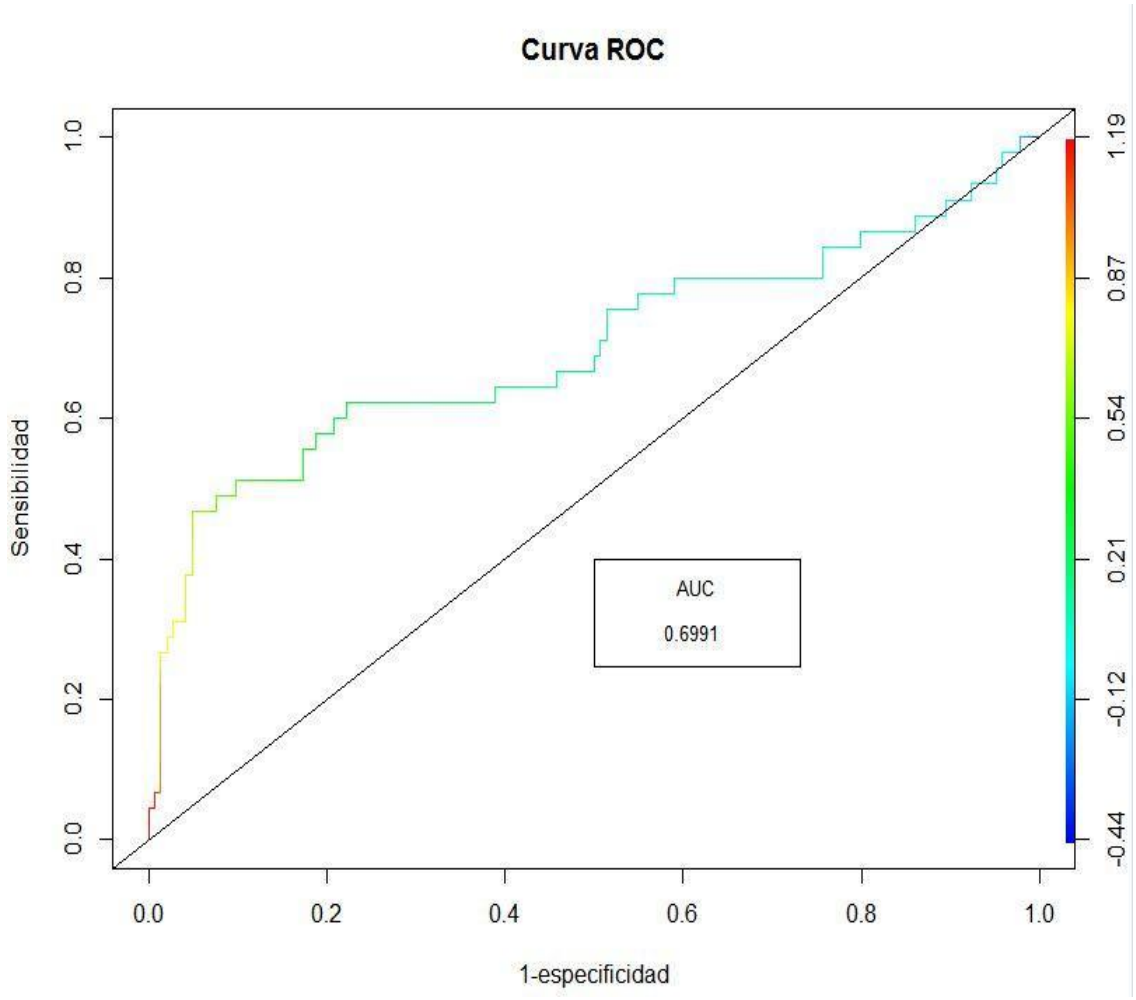


Figura 11. Análisis ROC modelo de red neuronal

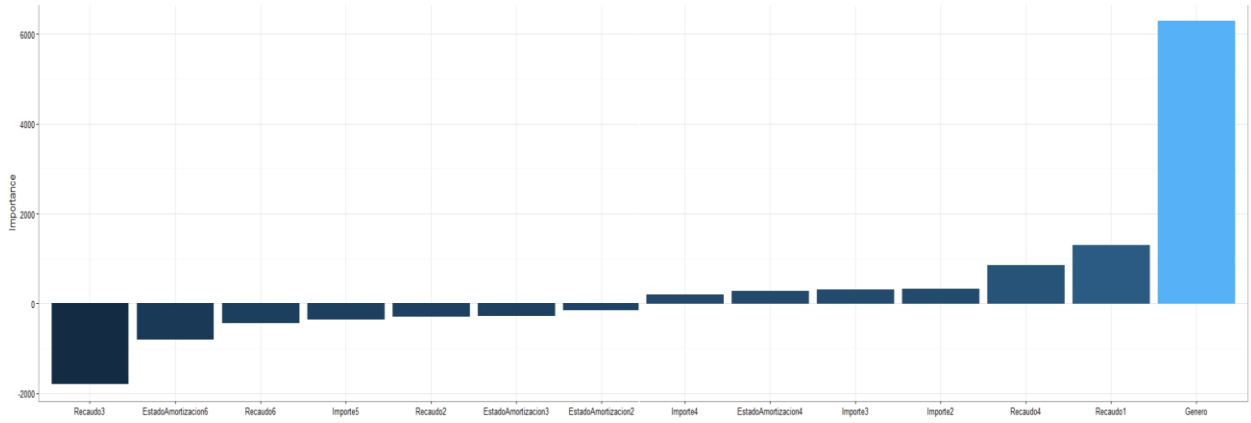


Figura 12. Análisis de sensibilidad modelo de red neuronal[Fn3]

5.3 Pruebas y resultados en modelo de árboles de decisión

5.3.1 Validación cruzada con diez iteraciones, evaluación entrenamiento y prueba y análisis ROC Algoritmo C4.5

Para la evaluación del modelo generado con el algoritmo C4.5 se utilizan dos tipos de pruebas: Validación cruzada con diez iteraciones para saber qué tan bien se ajustan los datos al modelo generado calculando la exactitud promedio y la prueba 90% y 10% para calcular los criterios de evaluación descritos en la sección 5.1. De igual manera se realiza un análisis ROC para saber la bondad del modelo y un análisis de sensibilidad para conocer el efecto o influencia de cada variable predictora sobre la variable dependiente. A continuación, se muestra los resultados obtenidos al aplicar validación cruzada con diez iteraciones sobre el conjunto de datos con el objetivo de obtener un promedio de la exactitud general del modelo y representar las variaciones entre cada iteración para saber si el modelo de árboles de decisión utilizando el algoritmo C4.5 se ajusta correctamente a los datos. En la Tabla 12 se muestra los resultados obtenidos en cada iteración. La validación cruzada obtuvo un promedio de exactitud de 73.68% con un error promedio de 24.5% con lo que se puede observar que el modelo se ajusta a los datos mejor que las redes neuronales.

Tabla 12. Validación cruzada 10 iteraciones modelo árboles de decisión algoritmo C4.5

Iteración	**Exactitud**
1	79.60%
2	77.19%
3	79.36%
4	78.57%
5	69.49%
6	75.51%
7	71.92%
8	85.18%
9	78.18%
10	72.22%

En la Tabla 13 se muestra los resultados obtenidos en la experimentación por el modelo de árboles de decisión con el algoritmo C4.5 utilizando el tipo de prueba 90% y 10%. Los parámetros calculados son la exactitud, la tasa de error, la sensibilidad, la especificidad, la precisión y el promedio de verdaderos negativos (NPV). El modelo seleccionado tiene un *minsplit* de 78 que aunque tiene una sensibilidad menor que el modelo con un *minsplit* de 150 se disminuye el riesgo de un sobre ajuste. La exactitud alcanzada por el modelo de árboles de decisión es de 0.8245 que indica que el 82.46% de las predicciones realizadas en los datos de prueba son correctas y por tanto la tasa de error es de 17.54% que indica el porcentaje de clientes mal clasificados. La sensibilidad que es la métrica más importante en este problema alcanzó un 64.28% que indica la probabilidad de que, dado un cliente que realmente incumple el pago el modelo lo detecte. Este resultado comparado con la especificidad que alcanzó 88.37% implica que el modelo detecta mejor los casos de clientes que no incumplen. Esto se debe a la disparidad significativa entre el número de casos de clientes que incumplen los pagos de sus cuotas y clientes que no incumplen los pagos de sus cuotas, la proporción es de 72% de casos que no incumplen las cuotas y 28% de casos que

incumplen las cuotas. La precisión alcanzada por el modelo es de 0.6428, es decir, cuando el modelo predice un cliente que incumplirá el pago de su cuota, acierta el 64,28% de las veces. El valor de predicción negativo alcanzó un 0.8837, es decir, que el 88.37% de los clientes que no incumplirán el pago fueron clasificados correctamente.

Tabla 13. Parámetros calculados a los modelos de árboles de decisión utilizando el algoritmo rpart.

Minsplit	Sensibilidad	Especificidad	Exactitud	Tasa de error	Precisión	NPV
20	0,357143	0,953488	0,807018	0,192982	0,714286	0,82
23	0,428571	0,930233	0,807018	0,192982	0,666667	0,833333
29	0,357143	0,976744	0,824561	0,175439	0,833333	0,823529
38	0,428571	0,953488	0,824561	0,175439	0,75	0,836735
41	0,571429	0,930233	0,842105	0,157895	0,727273	0,869565
43	0,5	0,930233	0,824561	0,175439	0,7	0,851064
47	0,5	0,906977	0,807018	0,192982	0,636364	0,847826
60	0,5	0,953488	0,842105	0,157895	0,777778	0,854167
78	0,642857	0,883721	0,824561	0,175439	0,642857	0,883721
86	0,642857	0,860465	0,807018	0,192982	0,6	0,880952
150	0,714286	0,813953	0,789474	0,210526	0,555556	0,897436

Al modelo de árboles de decisión utilizando el algoritmo C4.5 se le realiza un análisis ROC que muestra la representación gráfica de la sensibilidad del modelo frente a la especificidad. El parámetro área bajo la curva (AUC) se calcula para evaluar la bondad del modelo que alcanzó un 0.7824 que es considerado un test bueno y quiere decir que existe un 78.24% de probabilidad de que una clasificación realizado a un cliente que incumplirá el pago sea más correcto que el de un cliente que no incumplirá el pago escogido al azar. Este resultado supera al resultado obtenido por las redes neuronales. La Figura 13 muestra la curva ROC del modelo de árboles de decisión utilizando el algoritmo C4.5 con su parámetro AUC

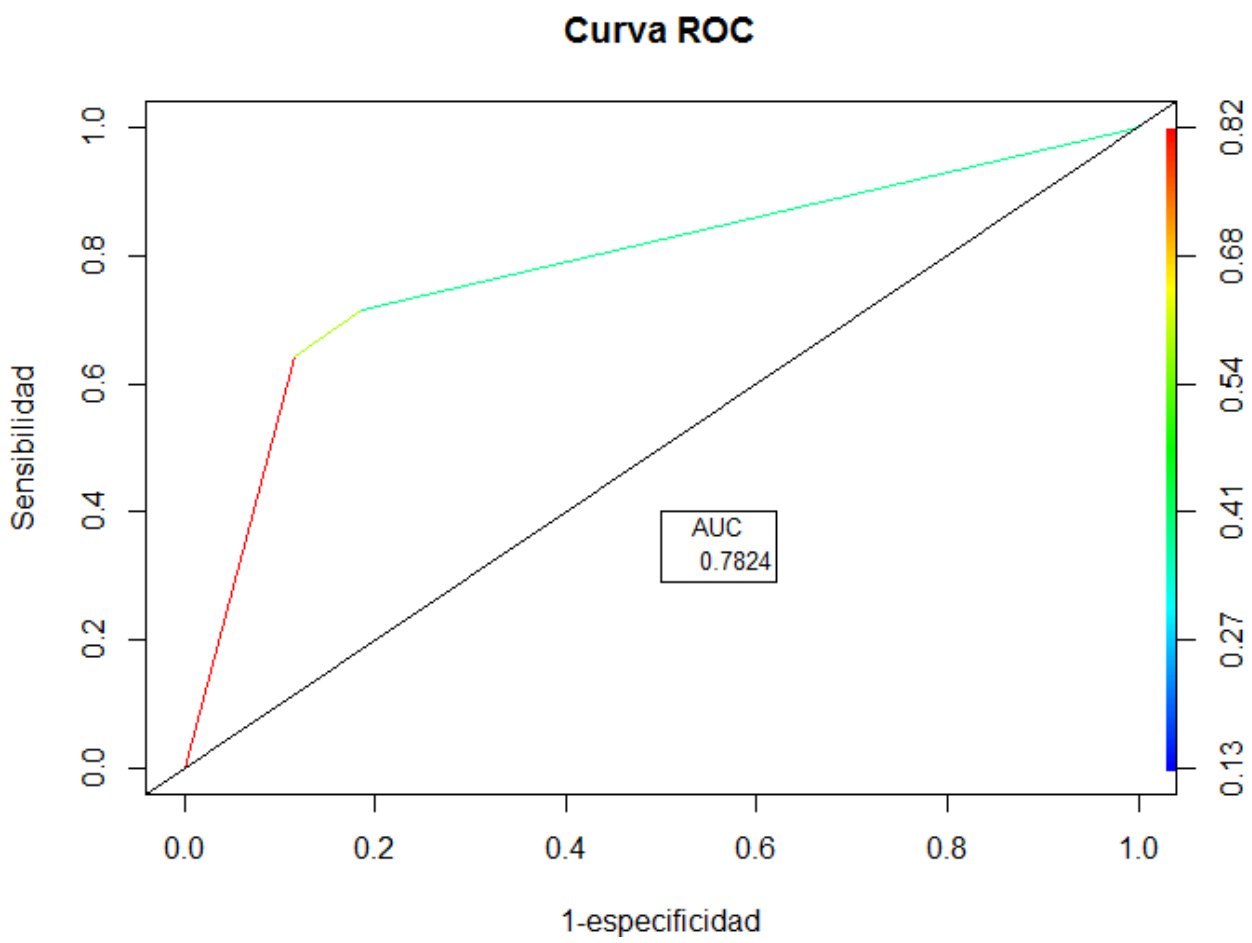


Figura 13. Análisis ROC modelo árboles de decisión Algoritmo C4.5

El análisis de sensibilidad se obtiene del resumen generado por el modelo utilizando la función summary en R. El análisis arroja que las variables Recaudo6 y EstadoAmortizacion6 tienen el porcentaje con mayor importancia con respecto a la variable dependiente. De igual manera las variables Recaudo3 e Importe5 tienen un porcentaje cercano a 0 que indica que tienen poca importancia para la variable dependiente. La Figura 14 muestra los porcentajes de importancia por cada variable independiente en el modelo de árboles de decisión utilizando el algoritmo C4.5. [Fn4]

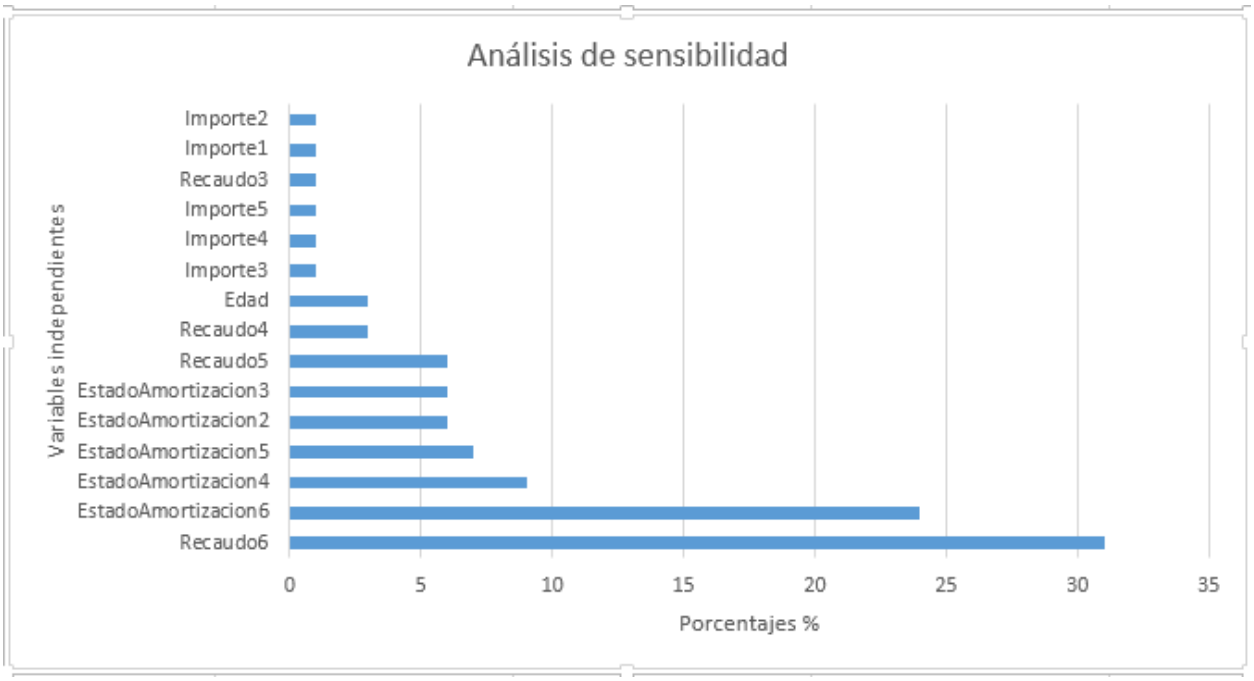


Figura 14. Análisis de sensibilidad algoritmo C4.5[Fn5]

5.3.2 Validación cruzada con diez iteraciones, evaluación entrenamiento y prueba y análisis ROC Algoritmo Random Forest

A continuación, se muestra los resultados obtenidos al aplicar validación cruzada con diez iteraciones sobre el conjunto de datos con el objetivo de obtener un promedio de la exactitud general del modelo y representar las variaciones entre cada iteración para saber si el modelo de árboles de decisión utilizando el algoritmo Random Forest se ajusta correctamente a los datos. En la validación cruzada se obtuvo un promedio de exactitud de 78.31% y un error promedio de 21.69%. Este modelo de árboles de decisión con el algoritmo Random Forest alcanza un mejor promedio de exactitud que el modelo de árboles de decisión con el algoritmo C4.5 y que el modelo de redes neuronales.

La Tabla 14 muestra los resultados obtenidos en la experimentación ajustando los hiperparámetros del algoritmo Random Forest utilizando el tipo de prueba 90% y 10%. Los parámetros calculados son la exactitud, la tasa de error, la sensibilidad, la especificidad, la precisión y el promedio de verdaderos negativos (NPV) con el respectivo parámetro de ajuste que es el *mtry* y *nodesize*. Los parámetros se seleccionan de acuerdo al modelo que tuvo el menor error que para el caso del *mtry* es 11. En la Tabla 15 se muestra los resultados obtenidos en la experimentación del hiperparámetro *nodeSize* cuyo valor que generó menor error es 14.

Tabla 14. Parámetros calculados a los modelos Random Forest ajustando el hiperparámetro *mtry*.

Mtry	oob_error rate	exactitud	Sensibilidad	Especificidad	Precisión	NPV
1	0.2757937	0.7242063	0.5106383	0.7461707	0.1714286	0.9368132
2	0.265873	0.734127	0.5416667	0.7662037	0.2785714	0.9093407
3	0.2519841	0.7480159	0.5783133	0.7814727	0.3428571	0.9038462
4	0.2420635	0.7579365	0.6	0.7922705	0.3857143	0.9010989
5	0.2361111	0.7638889	0.6129032	0.7980535	0.4071429	0.9010989
6	0.2321429	0.7678571	0.6161616	0.8049383	0.4357143	0.8956044
7	0.234127	0.765873	0.6170213	0.8	0.4142857	0.9010989
8	0.2400794	0.7599206	0.592233	0.8029925	0.4357143	0.8846154
9	0.2400794	0.7599206	0.5940594	0.8014888	0.4285714	0.8873626
10	0.2440476	0.7559524	0.5876289	0.7960688	0.4071429	0.8901099
11	0.2301587	0.7698413	0.6176471	0.8084577	0.45	0.8928571
12	0.2460317	0.7539683	0.5784314	0.7985075	0.4214286	0.8818681
13	0.2420635	0.7579365	0.5882353	0.800995	0.4285714	0.8846154
14	0.2380952	0.7619048	0.5980392	0.8034826	0.4357143	0.8873626
15	0.234127	0.765873	0.6078431	0.8059701	0.4428571	0.8901099
16	0.2361111	0.7638889	0.6060606	0.8024691	0.4285714	0.8928571
17	0.2301587	0.7698413	0.6176471	0.8084577	0.45	0.8928571
18	0.2361111	0.7638889	0.6039604	0.8039702	0.4357143	0.8901099
19	0.2361111	0.7638889	0.6039604	0.8039702	0.4357143	0.8901099
20	0.2440476	0.7559524	0.5841584	0.7990074	0.4214286	0.8846154
21	0.2480159	0.7519841	0.5742574	0.7965261	0.4142857	0.8818681
22	0.2380952	0.7619048	0.6041667	0.7990196	0.4142857	0.8956044
23	0.2440476	0.7559524	0.5841584	0.7990074	0.4214286	0.8846154
24	0.2321429	0.7678571	0.6161616	0.8049383	0.4357143	0.8956044

Tabla 15. Parámetros calculados a los modelos Random Forest ajustando el hiperparámetro *nodesize*.

Nodesize	oob_error rate	exactitud	Sensibilidad	Especificidad	Precisión	NPV
1	0.2380952	0.7619048	0.5961538	0.805	0.4428571	0.8846154
2	0.234127	0.765873	0.61	0.8044554	0.4357143	0.8928571
3	0.2420635	0.7579365	0.5957447	0.795122	0.4	0.8956044
4	0.2321429	0.7678571	0.6210526	0.801956	0.4214286	0.9010989
5	0.2321429	0.7678571	0.6236559	0.8004866	0.4142857	0.9038462
6	0.2301587	0.7698413	0.6276596	0.802439	0.4214286	0.9038462
7	0.2222222	0.7777778	0.6428571	0.8103448	0.45	0.9038462
8	0.2321429	0.7678571	0.6161616	0.8049383	0.4357143	0.8956044
9	0.2222222	0.7777778	0.6428571	0.8103448	0.45	0.9038462
10	0.2202381	0.7797619	0.6494845	0.8108108	0.45	0.9065934

11	0.218254	0.781746	0.65625	0.8112745	0.45	0.9093407
12	0.2242063	0.7757937	0.6363636	0.8098765	0.45	0.9010989
13	0.2242063	0.7757937	0.6336634	0.8114144	0.4571429	0.8983516
14	0.2202381	0.7797619	0.6494845	0.8108108	0.45	0.9065934
15	0.2162698	0.7837302	0.6631579	0.8117359	0.45	0.9120879
16	0.2242063	0.7757937	0.6336634	0.8114144	0.4571429	0.8983516
17	0.2242063	0.7757937	0.6363636	0.8098765	0.45	0.9010989
18	0.234127	0.765873	0.6122449	0.8029557	0.4285714	0.8956044
19	0.2261905	0.7738095	0.63	0.8094059	0.45	0.8983516
20	0.2242063	0.7757937	0.6391753	0.8083538	0.4428571	0.9038462

La exactitud alcanzada por el modelo con los parámetros seleccionados es de 0.8421 que indica que el 84.21% de las predicciones realizadas en los datos de prueba son correctas y por tanto la tasa de error es de 15.78% que indica el porcentaje de clientes mal clasificados. La sensibilidad que es la métrica más importante en este problema alcanzó un 57.14% que indica la probabilidad de que, dado un cliente que realmente incumple el pago el modelo lo detecte. Este resultado comparado con la especificidad que alcanzó 93.02% implica que el modelo detecta mejor los casos de clientes que no incumplen, esto se debe a la disparidad significativa entre el número de casos de clientes que incumplen los pagos de sus cuotas y clientes que no incumplen los pagos de sus cuotas. La precisión alcanzada por el modelo es de 0.7272, es decir, cuando el modelo predice un cliente que incumplirá el pago de su cuota, acierta el 72,72% de las veces. El valor de predicción negativo alcanzó un 0.8695, es decir, que el 86.95% de los clientes que no incumplirán el pago fueron clasificados correctamente.

Al modelo de árboles de decisión utilizando el algoritmo Random Forest se le realiza un análisis ROC que muestra la representación gráfica de la sensibilidad del modelo frente a la especificidad. El parámetro área bajo la curva (AUC) se calcula para evaluar la bondad del modelo que alcanzó un 88.29%. Este resultado supera al resultado obtenido por las redes neuronales y el algoritmo C4.5. La Figura 15 muestra la curva ROC del modelo de árboles de decisión utilizando el algoritmo Random Forest con su parámetro AUC

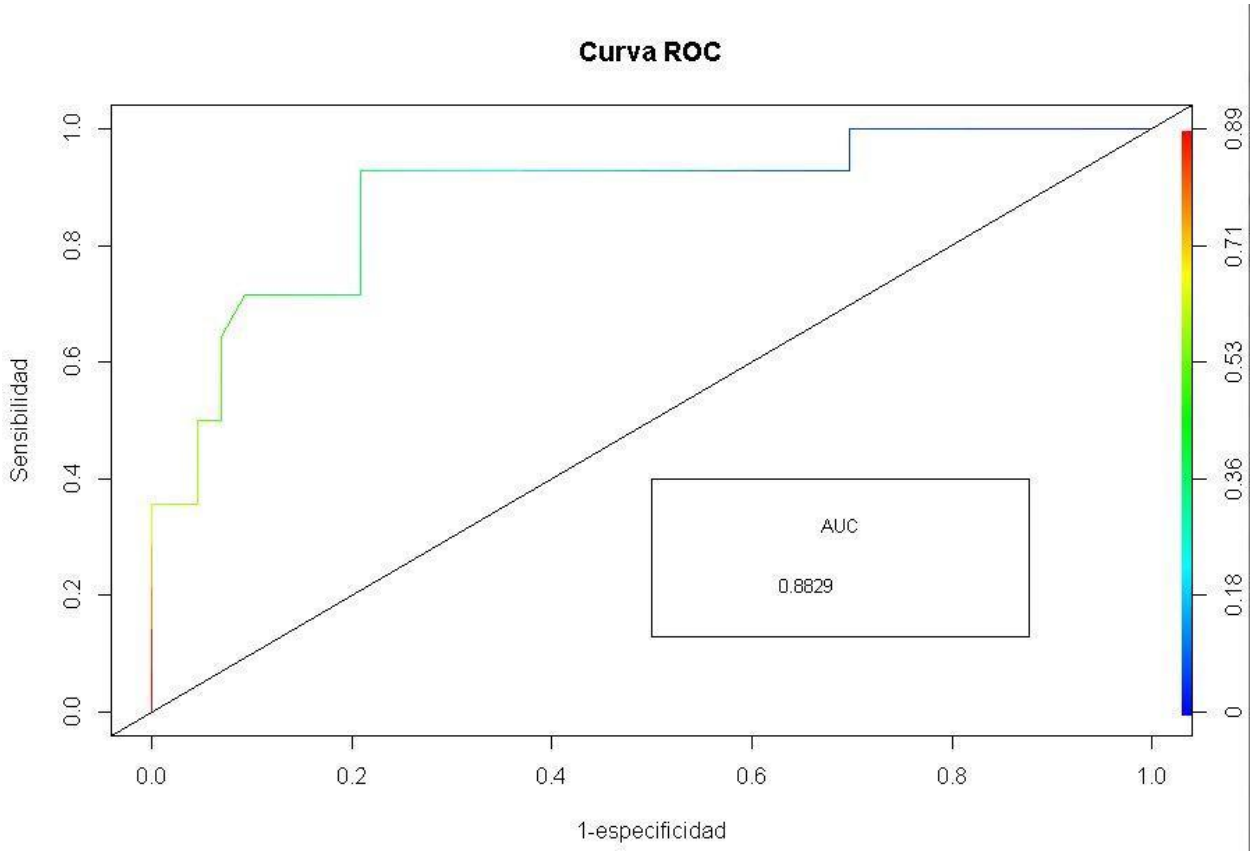


Figura 15. Análisis ROC modelo árboles de decisión Algoritmo Random Forest

El análisis de sensibilidad se obtiene de la función varImpPlot en R. El análisis arroja dos medidas el primero es la reducción de error medio (MDA) que indica el cambio de error al escoger al azar una variable y permutar las demás. El segundo es la reducción de la impureza (MDG) utilizando el criterio de Gini que es una medida de desorden. En las dos medidas variables Recaudo6 y EstadoAmortizacion6 tienen mayor importancia con respecto a la variable dependiente. De igual manera la variable Género tiene la menor importancia. La Figura 16 muestra los porcentajes de importancia por cada variable independiente en el modelo de árboles de decisión utilizando el algoritmo Random Forest.[Fn6]

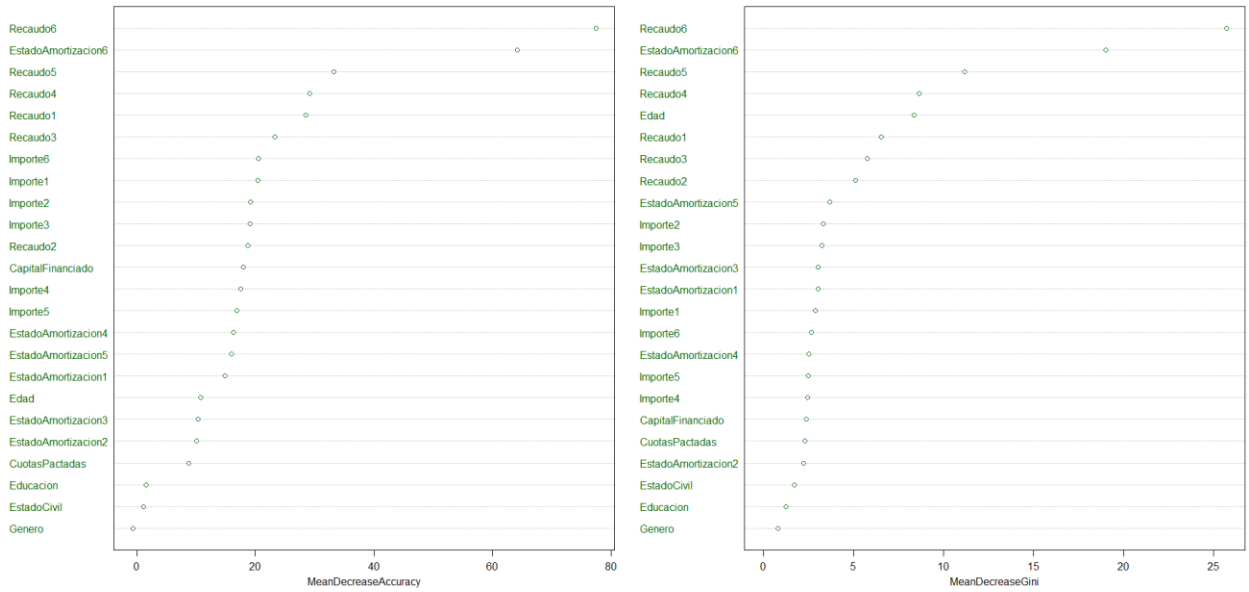


Figura 16. Análisis de sensibilidad algoritmo Random Forest[Fn7]

5.3.3 Validación cruzada con diez iteraciones, evaluación entrenamiento y prueba y análisis ROC Algoritmo C5.0

La exactitud del modelo obtenido usando el algoritmo C5.0 varía entre 52% y 80% en las diez iteraciones con un promedio de 64.93% y un error promedio de 35.07%. Este modelo de árboles de decisión con el algoritmo C5.0 alcanza un mejor promedio de exactitud que los modelos de redes neuronales y es superado por árboles de decisión con el algoritmo C4.5 y Random Forest.

En la Tabla 16 se muestra los resultados obtenidos por el modelo de árboles de decisión con el algoritmo C5.0 variando el costo que penaliza a los clientes que no pagan la cuota pactada y que el modelo predice que sí la pagan. Se utiliza el tipo de prueba 90% y 10%. Los parámetros calculados son la exactitud, la tasa de error, la sensibilidad, la especificidad, la precisión y el promedio de verdaderos negativos (NPV).

Tabla 16. Parámetros calculados a los modelos C5.0 ajustando el hiperparámetro *costs*.

Costs	Exactitud	Sensibilidad	Especificidad	Precisión	NPV	oob_error rate
1	0.8421053	0.3571429	1	1	08269231	0.15789474
2	0.754386	0.5	0.8372093	0.5	0.8372093	0.12280702
3	0.7368421	0.5	0.8139535	0.4666667	0.8333333	0.12280702
4	0.631500	0.6428571	0.4651163	0.28125	0.8	0.0877193
5	0.5614035	0.7142857	0.4883721	0.3125	0.84	0.07017544
6	0.5438596	0.8571429	0.4651163	0.3428571	0.9090909	0.03508772
7	0.4385965	0.8571429	0.3023256	0.2857143	0.8666667	0.03508772
8	0.4035088	0.8571429	0.255814	0.2727273	0.8461538	0.03508772
9	0.3508772	0.8571429	0.1860465	0.2553191	0.8	0.03508772

10	0.3508772	0.8571429	0.1860465	0.2553191	0.8	0.03508772

La exactitud alcanzada por el modelo de árboles de decisión utilizando el algoritmo C5.0 es del 63.15% y la tasa de error es del 8.77%. La sensibilidad alcanzó un 64.28%. Este resultado es superado por los modelos de las redes neuronales, iguala el algoritmo C4.5 y supera al algoritmo Random Forest. La especificidad, precisión, y precisión negativa fue de 46.51%, 28.12% y 80%, respectivamente. El área bajo la curva obtenida por el modelo de árboles de decisión utilizando el algoritmo C5.0 es del 80.56%. Este resultado supera al resultado obtenido las redes neuronales y el árbol de decisión con el algoritmo C4.5 e implica que el modelo de árboles de decisión utilizando el algoritmo C5.0 tiene una mejor exactitud que la red neuronal y el algoritmo C4.5. La Figura 17 muestra la curva ROC del modelo de árboles de decisión utilizando el algoritmo C5.0 con su parámetro AUC

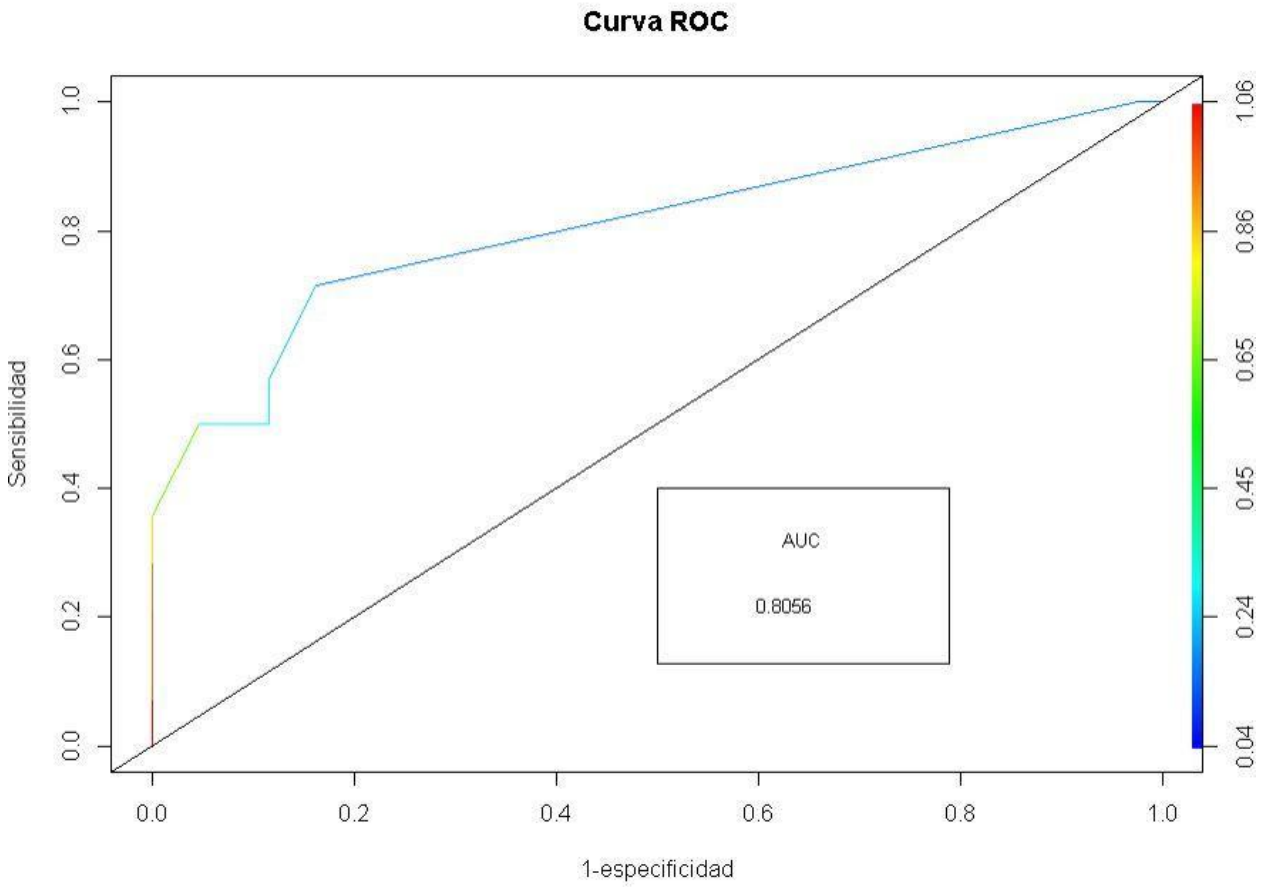


Figura 17. Análisis ROC modelo árboles de decisión Algoritmo C50

El análisis de sensibilidad se obtiene de la función C5imp en R. El análisis arroja que las variables EstadoAmortizacion6 y EstadoAmortizacion2 tienen el porcentaje con mayor importancia con respecto a la variable dependiente. De igual manera las variables Recaudo3 e Importe1 tienen un porcentaje cercano a 0 que indica que tienen poca importancia para la variable dependiente. La Figura 18 muestra los porcentajes de importancia por cada variable independiente en el modelo de árboles de decisión utilizando el algoritmo C5.0.[Fn8]

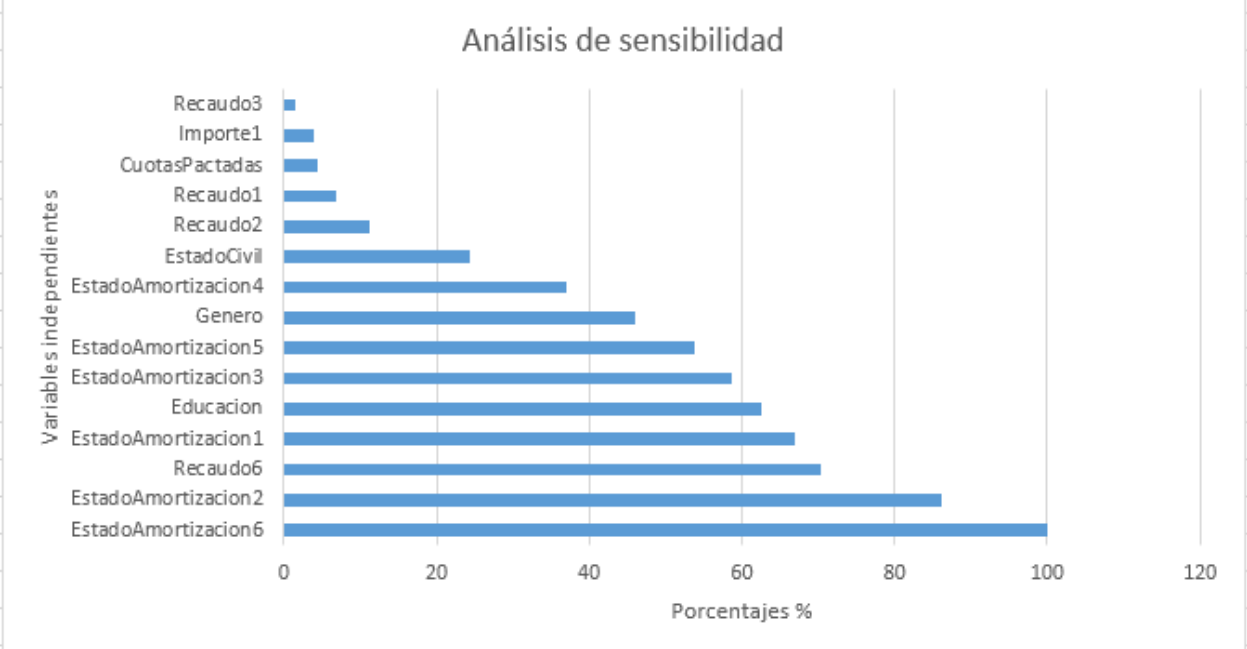


Figura 18. Análisis de sensibilidad modelo árboles de decisión Algoritmo C50[Fn9]

5.4 Pruebas y resultados usando Máquinas de Soporte Vectorial

El modelo obtenido con la técnica de máquinas de soporte vectorial alcanza una exactitud promedio de 60.32% y un error promedio de 39.68%.Este modelo es superado por árboles de decisión con el algoritmo C4.5, Random Forest, C5.0 y por la red neuronal de la Figura 6.El área bajo la curva obtenida por el modelo de máquinas de soporte vectorial es de 59.3%. Este resultado alcanzado por el modelo es el más bajo de todos e implica que el modelo no tiene una buena capacidad discriminatoria de clientes que incumplen el pago de su cuota y de los que no incumplen el pago de sus cuotas. La Figura 19 muestra la curva ROC del modelo de máquinas de soporte vectorial con su parámetro AUC

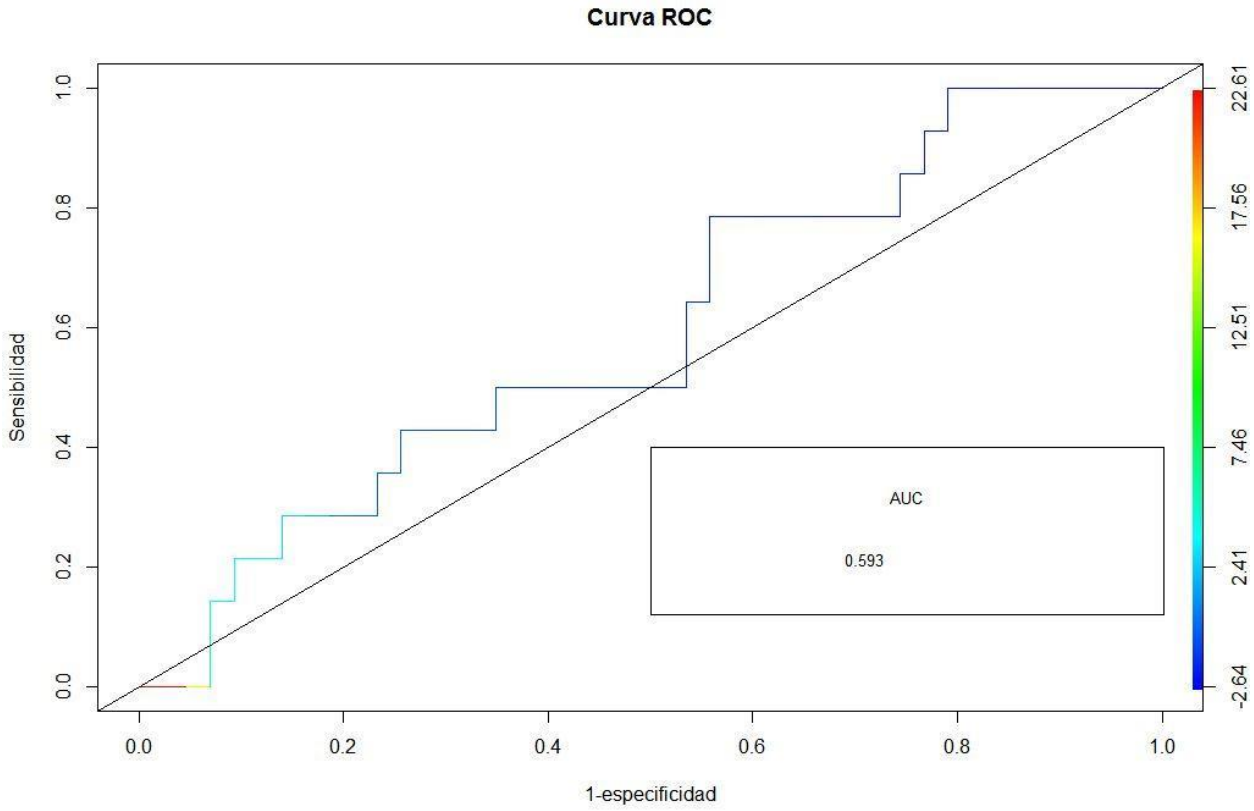


Figura 19. Análisis ROC modelo SVM

El análisis de sensibilidad se obtiene del resumen del modelo svm en R. El análisis arroja que las variables Recaudo6 y CuotasPactadas tienen el porcentaje con mayor importancia con respecto a la variable dependiente. De igual manera las variables EstadoAmortizacion5 y Genero tienen un porcentaje cercano a 0 que indica que tienen poca importancia para la variable dependiente. La Figura 20 muestra los porcentajes de importancia por cada variable independiente en el modelo de máquinas de soporte vectorial.[Fn10]

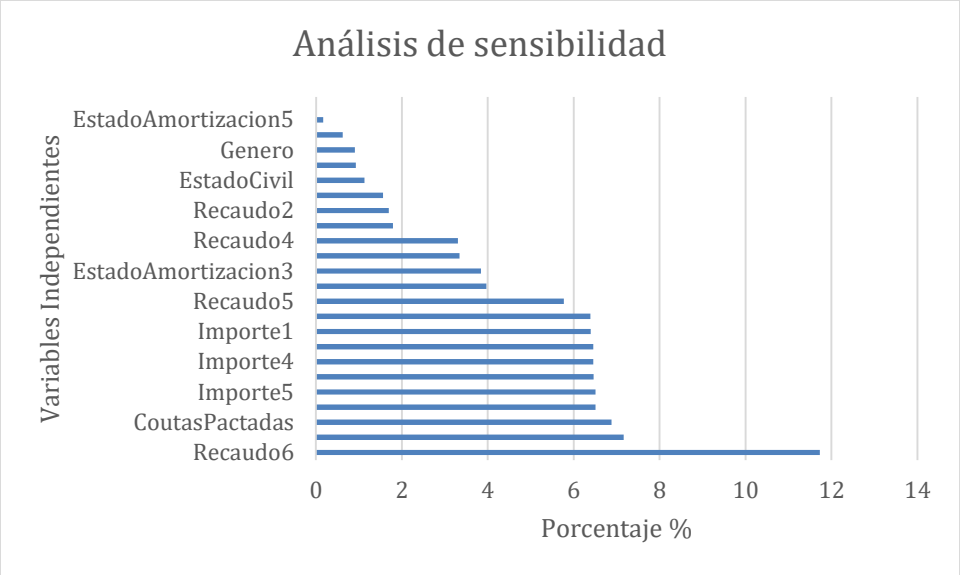


Figura 20. Análisis de sensibilidad modelo máquinas de soporte vectorial[Fn11]

La Tabla 17 muestra los resultados obtenidos por cada parámetro calculado por la función *tune* de la librería e1071 que permite realizar validación cruzada para ajustar los hiperparámetros *cost* y *gamma* por cada *kernel* del modelo de máquinas de soporte vectorial utilizando el tipo de prueba 90% y 10%. Los parámetros calculados son la exactitud, la tasa de error, la sensibilidad, la especificidad, la precisión y el promedio de verdaderos negativos (NPV).

Tabla 17. Resultados obtenidos por cada parámetro calculado por la función *tune*.

Cost	*gamma*	*kernel*	ssblidad	Esp.	Exactitud	Tasa de error	Precisión	NPV
1	0.042	lineal	0.1428	1	0.7894	0.2105	1	0.7818
0.125	0.125	Poly	0.2857	0.9534	0.7894	0.2105	0.6666	0.8039
1	0.125	radial	0.2142	1	0.8070	0.1929	1	0.7962
8	4	sigmoid	0.4444	0.8461	0.7192	0.2807	0.5714	0.7674

La exactitud y la tasa de error del modelo de máquinas de soporte vectorial fueron de 71.92% y 28.07%, respectivamente. La sensibilidad que es la métrica más importante en este problema alcanzó un 44.44% que indica la probabilidad de que, dado un cliente que realmente incumple el pago el modelo lo detecte. Este resultado comparado con la especificidad que alcanzó 84.61% indica que el modelo detecta mejor los casos en los que el cliente no incumplirá el pago de la cuota. Este resultado supera el modelo de árboles de decisión con el algoritmo C4.5 y es superado por Random Forest, C5.0 y las redes neuronales. Los porcentajes de precisión y precisión negativa son de 57.14% y 76.74%.

5.5 Análisis de resultados

El consolidado de las pruebas realizadas a los modelos se muestra en la Tabla 18.

Tabla 18. Resultados obtenidos de pruebas realizadas a los modelos generados.

Consolidado pruebas modelos de predicción								
Algoritmo	Sensibilidad	Especificidad	Exactitud	Tasa de error	Precisión	NPV	Prom. Exactitud	AUC
Modelo ANN	71.83	68.34	63.62	36.38	43.58	87.15	63.62	0.6991
C4.5	64.28	88.37	82.45	17.54	64.28	88.37	73.68	0.7824
Random Forest	57.14	93.02	84.21	15.78	72.72	86.95	78.31	0.8829
C5.0	64.28	46.51	63.15	8.77	28.12	80	64.93	0.8056
SVM	44.44	84.61	71.92	28.07	57.14	76.74	60.32	0.593

De los resultados obtenidos se puede observar que el modelo que obtuvo un mejor promedio de exactitud utilizando validación cruzada de diez iteraciones es el árbol de decisión que implementa el algoritmo Random Forest. De igual manera, el algoritmo Random Forest alcanza el mayor porcentaje de área bajo la curva que refleja un buen rendimiento del modelo. Sin embargo, la mejor sensibilidad la obtuvo el modelo de árboles de decisión que alcanzó un 83.33% seguido por el modelo de Redes Neuronales que alcanzó un 75.55%. Para el problema que se aborda en este trabajo de grado el parámetro más importante es la sensibilidad ya que realizar una predicción que clasifique al cliente como un cliente que no va a incumplir y termine incumpliendo el pago de la cuota es más costoso que un cliente que se clasifique como un cliente que incumplirá y que cumpla con el pago.

De acuerdo a los resultados obtenidos por todos los modelos implementados se desea saber qué modelo tiene el mejor rendimiento y qué modelo generaliza mejor el comportamiento observado. Para esto se realiza un ranking de los modelos evaluando el resultado del parámetro AUC ordenándolos de mayor a menor ya que cuanto más se acerque este valor a 1, mejor será el rendimiento del modelo (es decir, se maximiza la tasa de verdaderos positivos a la vez que se minimiza la de falsos positivos). También se realiza un ranking del parámetro promedio exactitud generado a partir de la validación cruzada de 10 iteraciones en cada modelo ordenándolas de mayor a menor con el objetivo de determinar el nivel en que los modelos se podrían generalizar para nuevos conjuntos de datos. La Tabla 19 muestra el ranking de los modelos por AUC (Área bajo la curva).

Tabla 19 Ranking de los modelos por AUC

Ranking modelos por AUC	
Algoritmo	AUC (Área bajo la curva)
Random Forest	0.8829
C5.0	0.8056
C.45	0.7973
Modelo ANN	0.6991
SVM	0.593

La Tabla 20 muestra el ranking de los modelos por promedio de exactitud validación cruzada de 10 iteraciones.

Tabla 20 Ranking de los modelos por promedio de exactitud validación cruzada de 10 iteraciones.

Ranking modelos por promedio de exactitud	
Algoritmo	Validación cruzada K=10
Random Forest	78.31%
C4.5	75.7%
C5.0	64.93%
Modelo ANN	63.62%
SVM	60.32%

En los dos rankings el mejor modelo es árboles de decisión con el algoritmo Random Forest seguido del algoritmo C5.0 y C4.5. El valor de exactitud más bajo lo obtuvo el modelo de máquinas de soporte vectorial que indica un modelo que tiende a realizar predicciones que se acerca a una estimación pseudoaleatoria.

En el análisis de sensibilidad generado en cada modelo se observa que los tres algoritmos utilizados en arboles de decisión (C4.5, Random Forest, C5.0) comparten las mismas variables de mayor importancia que son el Recaudo6 y el EstadoAmortizacion6. De igual manera el algoritmo de máquinas de soporte vectorial genera como variables de mayor importancia el recaudo 6 y las cuotas pactadas mientras que en las redes neuronales las variables independientes de mayor importancia relativa son el Género y el recaudo 3.

Comparando los resultados obtenidos con los antecedentes encontrados en el estado del arte como el trabajo presentado en (Turkson et al., 2016) donde emplearon 15 algoritmos de aprendizaje supervisado sobre un conjunto de datos de 30000 registros alcanzaron un promedio de exactitud entre el 76% y el 80% que indica que algunos algoritmos que se emplearon en este trabajo de grado están en el rango esperado como lo son los algoritmos de árboles de decisión. Por otra parte, el algoritmo que presentó el peor rendimiento como lo es el SVM está muy por debajo del resultado esperado.

CAPÍTULO 6

Prototipo de sistema para análisis de riesgo crediticio

En este capítulo se presentan los artefactos generados en el proceso de desarrollo del prototipo web que está compuesto por dos proyectos Back-end y Front-end. El Back-end está desarrollado en la plataforma .NET Core con el lenguaje C# que proporciona una API REST con autenticación por token. También se utiliza Entity Framework para mapear la base de datos y la librería R.NET para la interoperabilidad de R con C#. El Front-end está desarrollado en Angular versión 4. Para el desarrollo se utiliza el Framework Boilerplate que permite construir aplicaciones y sitios web responsive.

6.1 Requerimientos del prototipo

A continuación, se presenta los requerimientos generados para desarrollar el prototipo de Gestión de Riesgo Crediticio

Tabla 21 requerimientos generados para desarrollar el prototipo de Gestión de Riesgo Crediticio.

Programa	Descripción	Tareas
Usuarios del prototipo Gestión de riesgo crediticio	Mediante este programa, el usuario que tenga permisos de administrador podrá realizar las operaciones de crear, consultar, modificar, asignar contraseña, y borrar los diferentes usuarios que interactuaran con el sistema.	Consultar los usuarios previamente creados en el sistema.
		Al crear un usuario se permitirá capturar login, nombre, correo electrónico, tipo de rol (Administrador, Operario), estado (por defecto activo).
		Se permitirá eliminar un usuario, el sistema emitirá un mensaje de confirmación y no se podrá eliminar del sistema un usuario si no se cuenta con un rol de administrador
Roles del Prototipo	Mediante este programa, el usuario administrador podrá realizar las operaciones de crear, consultar, modificar y borrar los diferentes roles del sistema.	Consultar los roles previamente creados en el sistema.
		Al crear un rol se permitirá capturar la descripción de este.
		Al editar un perfil se permitirá modificar la descripción y el estado (Activo, Inactivo).
		Se permitirá eliminar un role, el sistema emitirá un mensaje de confirmación y no se podrá eliminar del sistema un role si no se tiene los permisos necesarios.
		Consultar los usuarios previamente creados en el sistema.

Asociación de usuarios y roles del sistema	Mediante este programa el usuario podrá realizar la asignación del role que el usuario tendrá en el sistema.	Al seleccionar un usuario el sistema permitirá asignar un role al usuario.
Asociación de permisos del sistema y roles del sistema	Mediante este programa el usuario podrá realizar la asignación de los permisos del sistema a los que el role tendrá en el sistema.	Consultar los roles del sistema previamente creados. Al seleccionar un role el sistema desplegará una lista de los permisos disponibles para asignar al role.
Login o inicio de sesión	Mediante este programa el usuario podrá realizar el inicio de sesión al sistema.	Validar que el inicio de sesión sea correcto. Bloquear el inicio de sesión del usuario por N de intentos fallidos.
módulo de análisis de riesgo crediticio	Mediante este programa, el usuario podrá realizar predicciones del pago de la próxima cuota pactada de un cliente ingresando los datos requeridos por el modelo de predicción (Capital Financiado, Genero, Nivel Educativo, Estado Civil, Edad, Últimos seis estados de amortización, importe de estado de cuenta de los últimos seis pagos y monto pagado) así como el id del cliente. También permitirá realizar una carga masiva de los datos para realizar predicciones masivas y seleccionar el tipo de modelo para realizar la predicción.	Para realizar una predicción el sistema permitirá al usuario escoger el modelo que desea utilizar en la predicción (Redes Neuronales, Árboles de decisión, Máquinas de soporte vectorial) también permitirá ingresar los datos esperados por el modelo de forma manual o de forma masiva por medio de una plantilla de Excel. El sistema generará un reporte que brindará información de las predicciones generadas por el modelo.

6.2 Historias de usuario

A continuación, se presentan las historias de usuario generadas por cada funcionalidad en los módulos de seguridad y gestión crediticia

Formato de requisitos		Versión
		1
		Página 1 de 1

Fecha:	01 sep 2018			
Requerimiento:	1			
Cliente:	Proyecto Trabajo de Grado		**Proyecto:**	GestionCrediticia 1.0

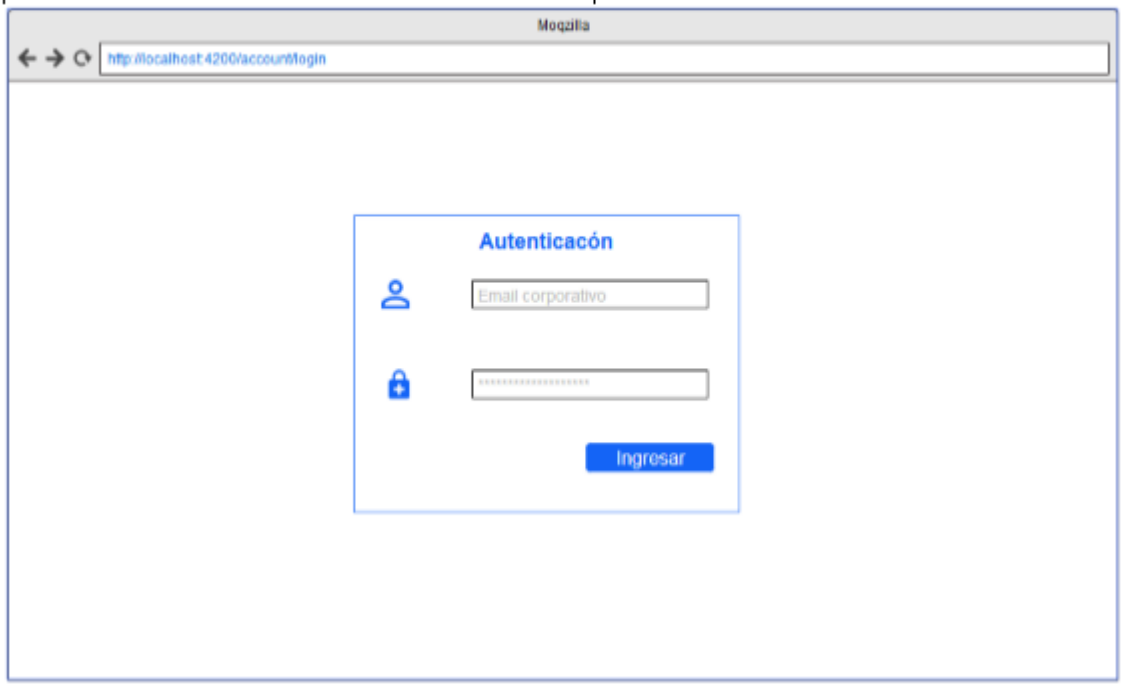
Módulo / Programa:
Inicio se sesión

Nombre del requisito:
El sistema permitirá el inicio de sesión

Descripción:
Este programa es el primero que se ejecuta en el sistema, y determina si el usuario podrá ingresar a partir de su login y contraseña digitada.

Excepciones y/o validaciones:
• Validar el que inicio de sesión sea correcto. • Bloquear el inicio de sesión del usuario por N de intentos fallidos.

Responsable:	Diego Andres Borrero
Firma:	



Formato de requisitos		Versión
		1
		Página 1 de 1

Fecha:	01 sep 2018			
Requerimiento:	2			
Cliente:	Proyecto Trabajo de Grado		**Proyecto:**	GestionCrediticia 1.0

Módulo / Programa:
Seguridad => Usuarios.

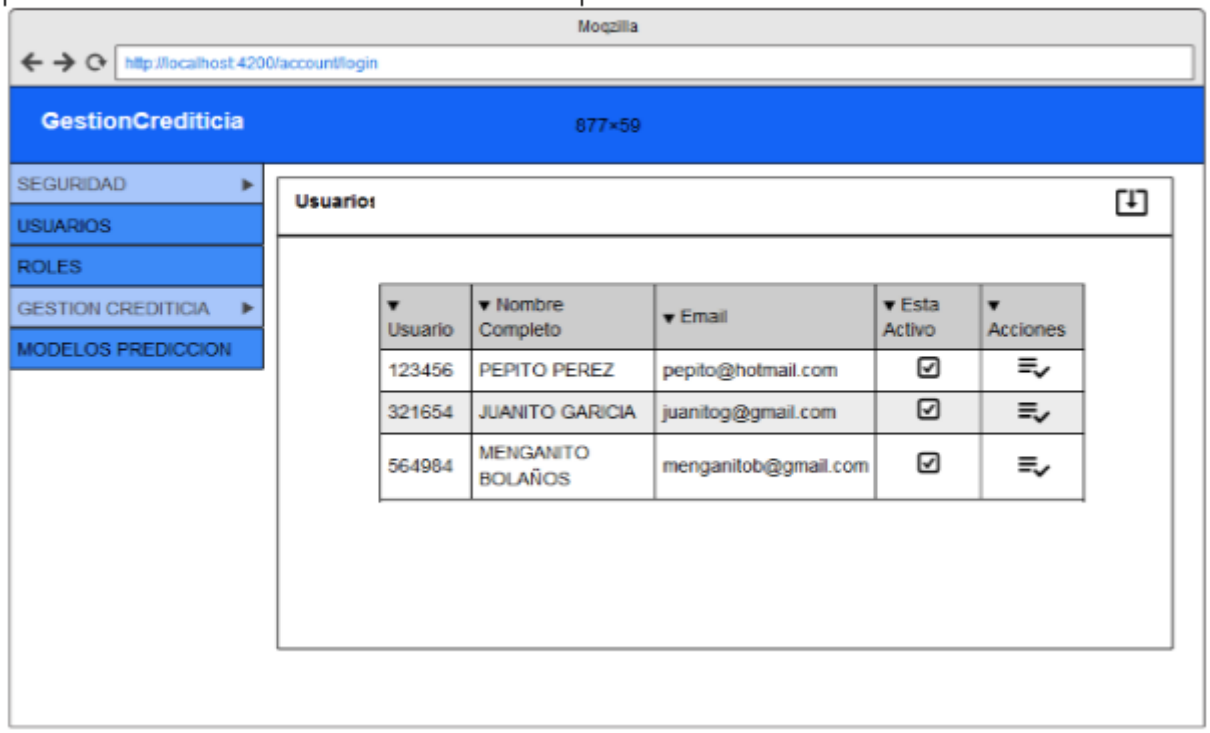
Nombre del requisito:
El sistema permitirá consultar los usuarios del sistema

Descripción:
En la opción **Seguridad => Usuarios,** el sistema desplegará una lista de los usuarios previamente creados en el sistema con las columnas: usuario, nombre completo, email, estado y las acciones (editar y eliminar).

Excepciones y/o validaciones:
Las acciones de modificar o eliminar usuario serán habilitadas si el usuario tiene los permisos para realizarlas.

Responsable: Diego Andres Borrero

Firma:



Formato de requisitos		Versión
		1
		Página 1 de 1

Fecha:	01 sep 2018			
Requerimiento:	3			
Cliente:	Proyecto Trabajo de Grado		**Proyecto:**	GestionCrediticia 1.0

Módulo / Programa:

Seguridad => Usuarios => Agregar usuario.

Nombre del requisito:

El sistema permitirá agregar un nuevo usuario del sistema.

Descripción:

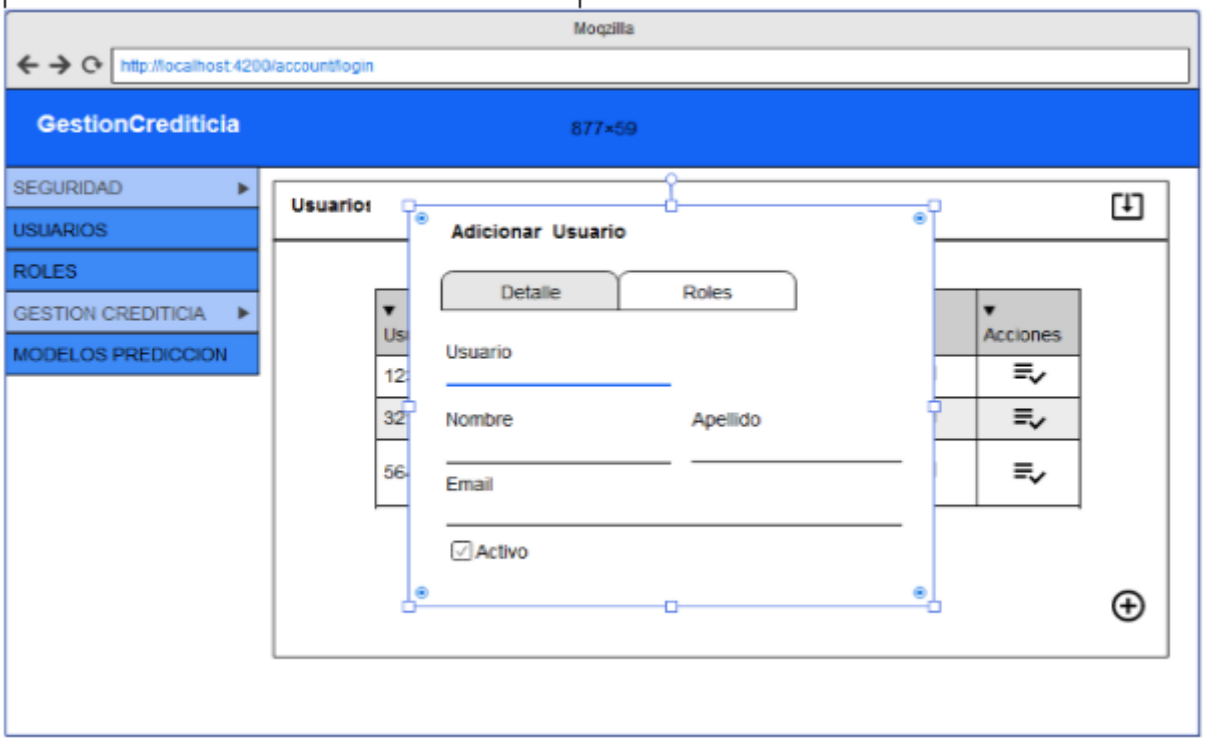
En la opción **Seguridad => Usuarios => Agregar usuario,** el sistema desplegará un formulario que permitirá capturar el usuario, el nombre, apellido, correo electrónico el estado estará por defecto en activo.

Excepciones y/o validaciones:

El sistema validará que el login ingresado no se encuentre registrado en el sistema.

Responsable: Diego Andres Borrero

Firma:



Formato de requisitos		Versión
		1
		Página 1 de 1

Fecha:	01 sep 2018			
Requerimiento:	**4**			
Cliente:	Proyecto Trabajo de Grado		**Proyecto:**	GestionCrediticia 1.0

Módulo / Programa:
Seguridad => Usuarios => Modificar o consultar usuario.

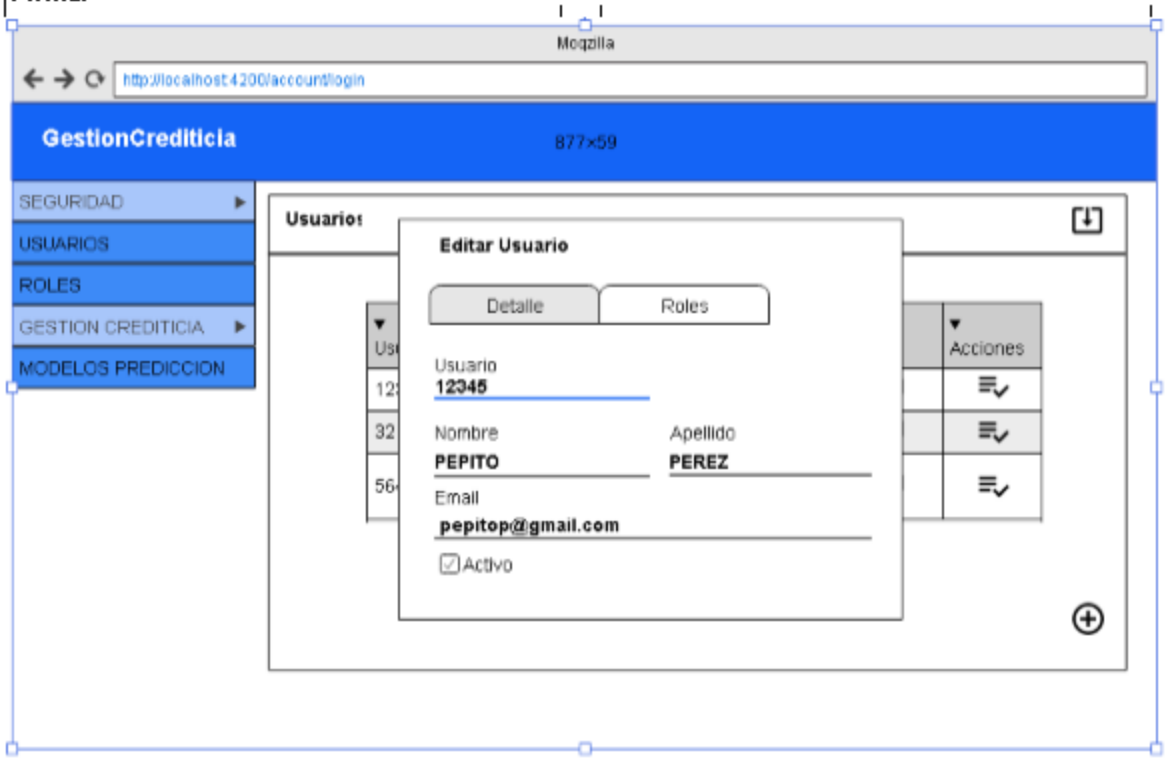
Nombre del requisito:
El sistema permitirá modificar o consultar un usuario del sistema.

Descripción:
En la opción **Seguridad => Usuarios => Editar,** el sistema desplegará un formulario que permitirá modificar el nombre,apellido, el correo electrónico, el rol (Administrador, Operario) y el estado (activo o inactivo).

Excepciones y/o validaciones:
El sistema validará si el usuario tiene permiso para modificar la información de los usuarios, si no la tiene solo le permitirá consultar la información.

Responsable: Diego Andres Borrero

Firma:



Formato de requisitos		Versión
		1
		Página 1 de 1

Fecha:	01 sep 2018			
Requerimiento:	**5**			
Cliente:	Proyecto Trabajo de Grado		**Proyecto:**	GestionCrediticia 1.0

Módulo / Programa:

Seguridad => Usuarios => Eliminar usuario.

Nombre del requisito:

El sistema permitirá eliminar un usuario del sistema.

Descripción:

En la opción **Seguridad => Usuarios => Eliminar usuario,** el sistema emitirá un mensaje de confirmación si el usuario desea eliminar el usuario, al confirmarse se eliminará el usuario seleccionado.

Excepciones y/o validaciones:

En el sistema no se podrá eliminar del sistema un usuario que tenga registros asociados.

Responsable: Diego Andres Borrero

Firma:

Formato de requisitos		Versión
		1
		Página 1 de 1

Fecha:	01 sep 2018			
Requerimiento:	6			
Cliente:	Proyecto Trabajo de Grado		**Proyecto:**	GestionCrediticia 1.0

Módulo / Programa:

Seguridad => Perfiles.

Nombre del requisito:

El sistema permitirá consultar los roles del sistema

Descripción:

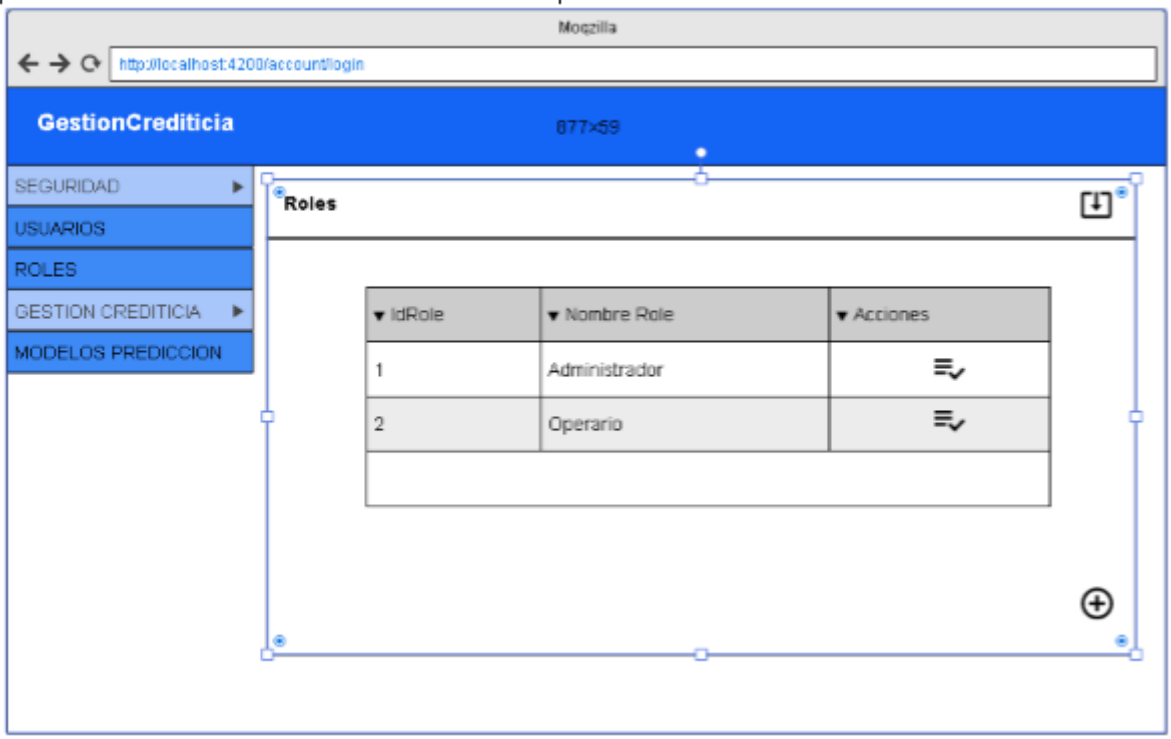
En la opción **Seguridad => Roles** el sistema desplegará una lista de los Roles previamente creados en el sistema con las columnas: Id Role, Nombre Role y las acciones (Editar y Eliminar) el sistema permitirá agregar roles

Excepciones y/o validaciones:

Las acciones de modificar o consultar, eliminar o agregar role serán habilitadas si el usuario tiene los permisos para realizarlas.

Responsable: Diego Andres Borrero

Firma:



Formato de requisitos		Versión
		1
		Página 1 de 1

Fecha:	01 sep 2018			
Requerimiento:	7			
Cliente:	Proyecto Trabajo de Grado		**Proyecto:**	GestionCrediticia 1.0

Módulo / Programa:

Seguridad => Perfiles => Agregar perfil.

Nombre del requisito:

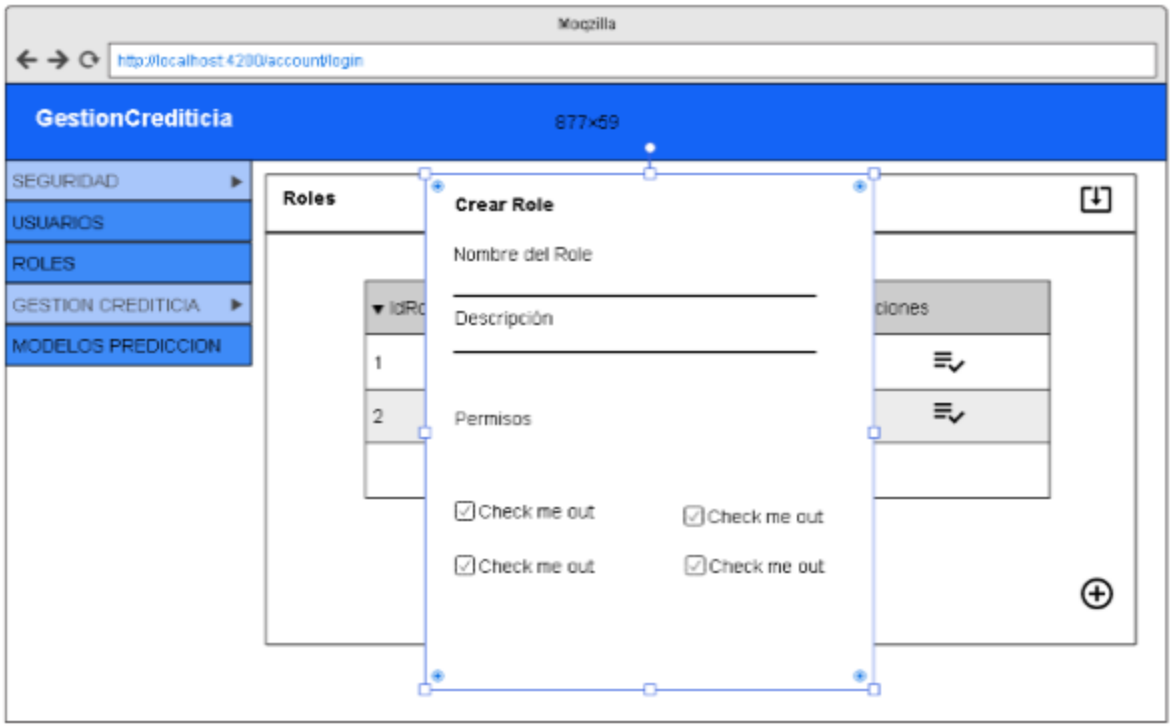
El sistema permitirá agregar un nuevo perfil del sistema

Descripción:

En la opción **Seguridad => Perfiles => Agregar role,** el sistema desplegará un formulario que permitirá capturar el código y la descripción; el estado estará por defecto en activo.

Excepciones y/o validaciones:

Responsable:	Diego Andres Borrero
Firma:	



Formato de requisitos		Versión
		1
		Página 1 de 1

Fecha:	01 sep 2018			
Requerimiento:	7			
Cliente:	Proyecto Trabajo de Grado		**Proyecto:**	GestionCrediticia 1.0

Módulo / Programa:
Gestion Crediticia

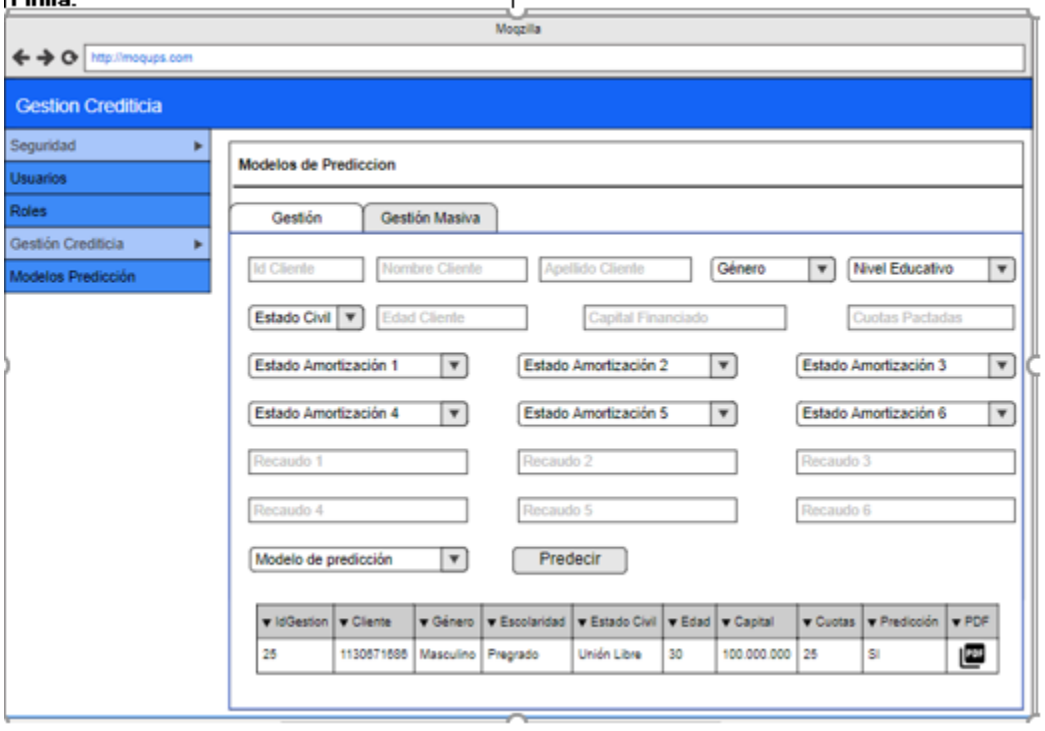
Nombre del requisito:
El sistema utilizar los modelos de prediccion generados en R para predecir incumplimientos de clientes

Descripción:
En la opción **Gestión Crediticia => Modelos Predicción => Gestión**, el sistema desplegará un formulario que permitirá capturar los datos necesario para generar una predicción según el modelo seleccionado.

Excepciones y/o validaciones:

Responsable: Diego Andres Borrero

Firma:



6.3 Modelo de datos

El modelo de datos para el prototipo de gestión crediticia se implementó utilizando el motor de bases de datos SQL Server 2014 Express. El modelo de datos se muestra en dos módulos llamados Seguridad y Gestión Crediticia y cada tabla cuenta con una llave primaria auto incrementable.

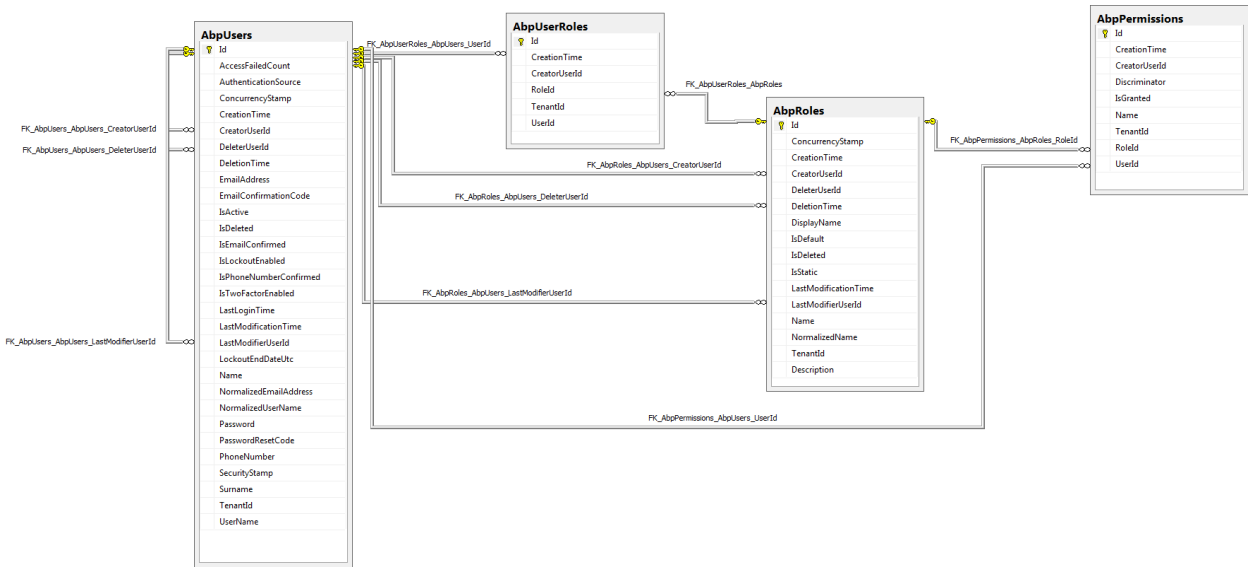


Figura 21. MER del módulo de seguridad

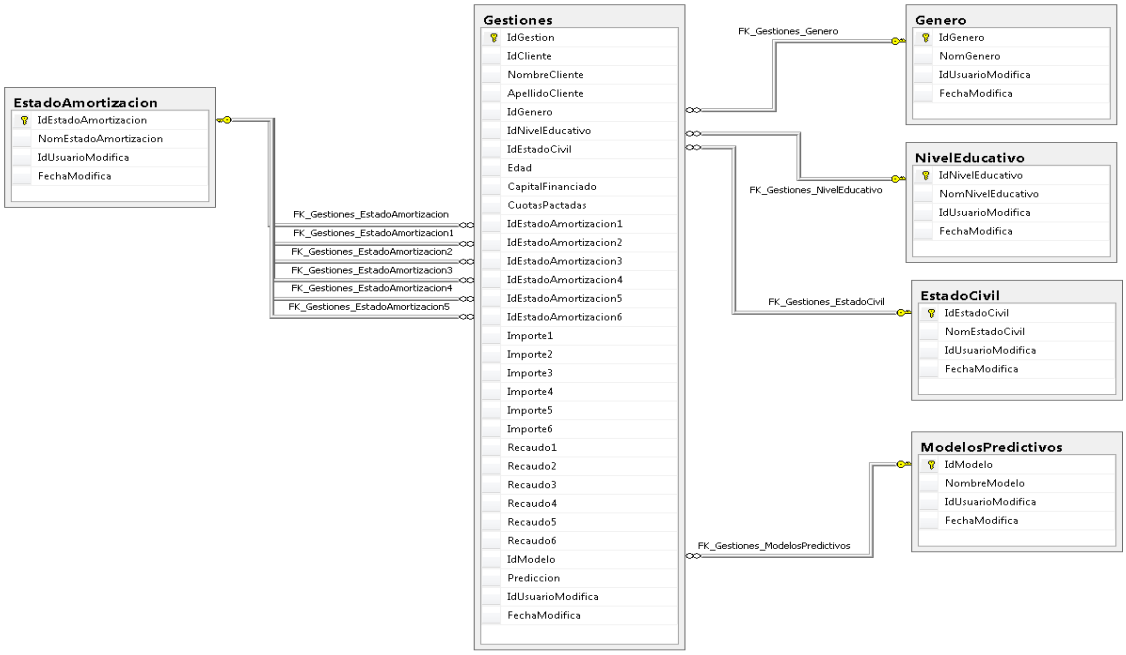


Figura 22. MER del Módulo Gestión Crediticia

6.4 Diagrama de clases

A continuación, se presenta el diagrama de clases implementado con Visual Studio 2017. Las clases generadas del módulo de seguridad vienen implementadas del framework Boilerplate con las cuales se realiza la administración de usuarios en el prototipo de Gestión Crediticia. Las clases identity del módulo de gestión Crediticia son generadas a partir del ORM EntityFramework 4.5 que son (Gestiones, Genero, EstadoCivil, NivelEducativo, Importes, EstadoAmortizacion y ModelosPredictivos). Para cada una de estas clases identity se generó una clase Manager que implementa los métodos para obtener, insertar y procesar información de la gestión crediticia. La Figura 23 muestra el diagrama de clases del módulo seguridad y cuenta con las clases (AbpUsers,AbpRoles y AbpPermission) y la Figura 24 muestra el diagrama de clases del módulo de gestiones que cuenta con las clases (Gestiones, GestionesManager, Genero, GeneroManager ,EstadoCivil, EstadoCivilManager, NivelEducativo, NivelEducativoManager, Importes, ImportesManager, EstadoAmortizacion, EstadoAmortizacionManager, ModelosPredictivos y ModelosPredictivosManager).

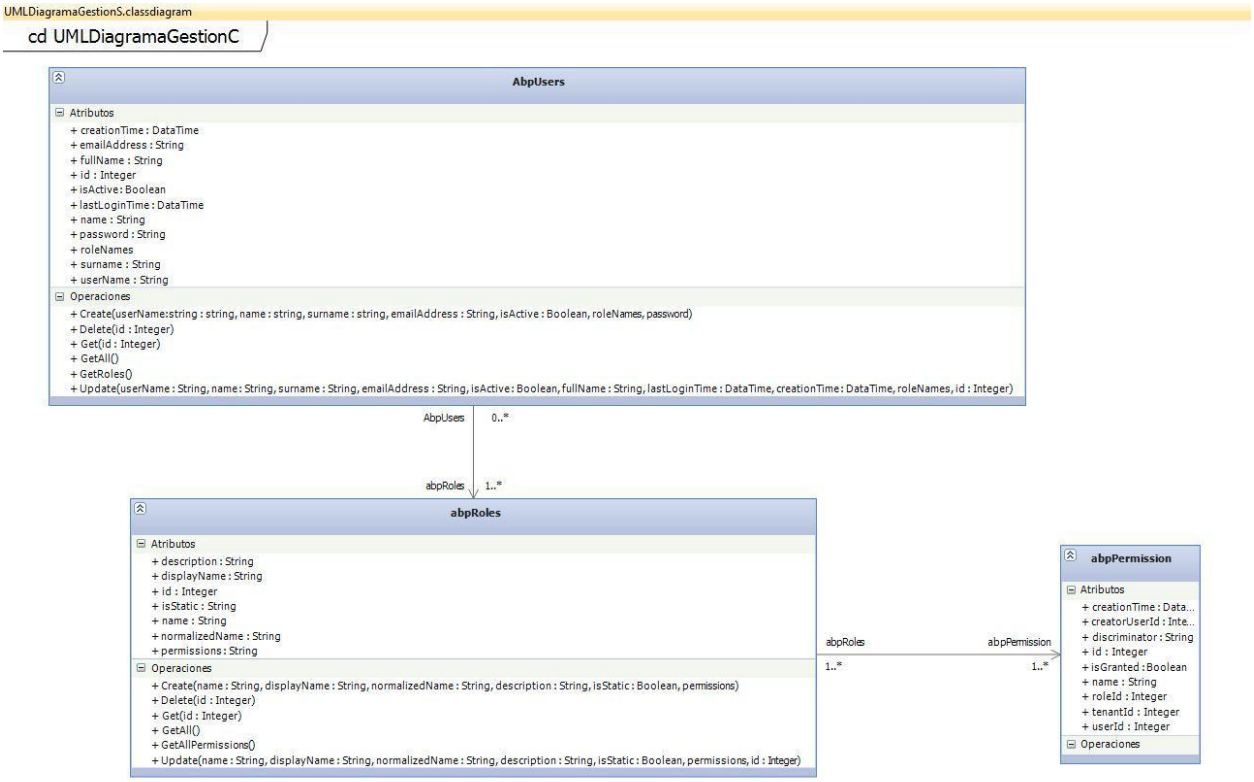


Figura 23. Diagrama de clases del módulo de seguridad

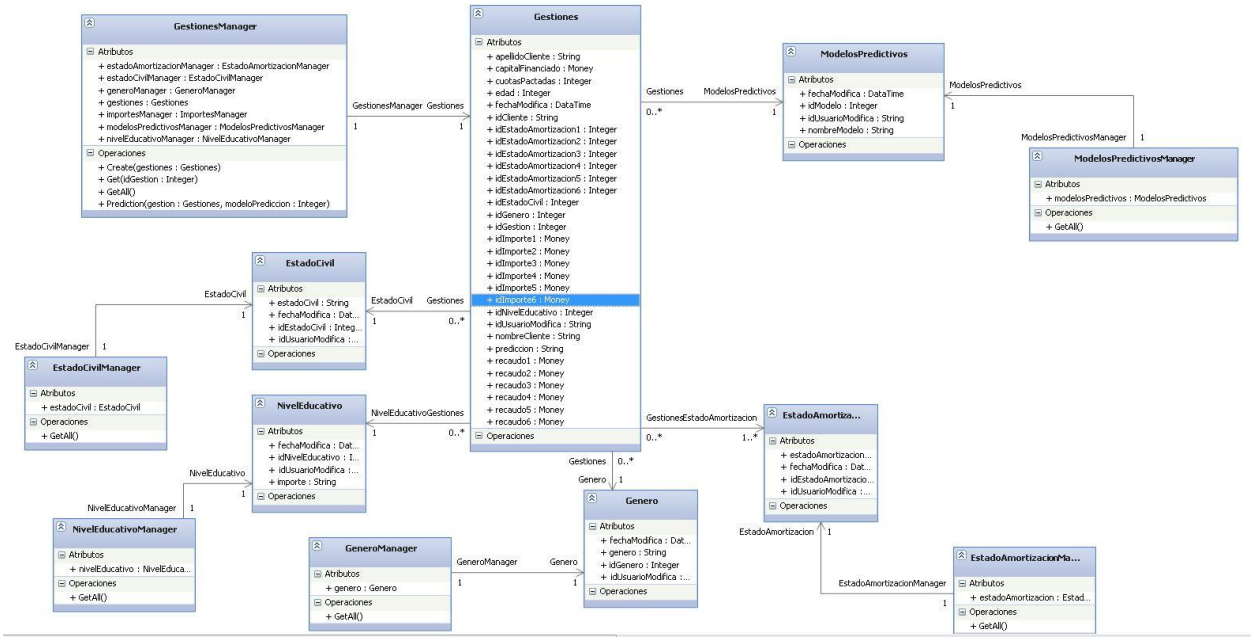


Figura 24. Diagrama de clases del módulo de gestión crediticia

6.5 Descripción de los módulos

Para implementar el prototipo de gestión crediticia que utilice los modelos de aprendizaje supervisado planteados en este documento se generaron dos módulos:

Módulo de seguridad: Este módulo permite la administración de usuarios que van a interactuar con el prototipo Gestión Crediticia donde permitirá la creación de usuarios, roles y permisos. Cada usuario puede tener uno o más roles los cuales tienen asignados una serie de permisos dependiendo del perfil del usuario.

Módulo Gestión: Este módulo permite la administración y generación de predicciones ya sea individual o masiva la cual cuenta con un reporte informativo. Las predicciones serán generadas de acuerdo al modelo de aprendizaje seleccionado por el usuario y brindará información de las aptitudes alcanzadas por cada modelo descritas en este documento (Sensibilidad, Especificidad, Exactitud, Tasa de error, NPV, Precisión y AUC).

6.6 Módulo de seguridad

Para que el usuario del área de gestión crediticia pueda ingresar al sistema debe digitar su usuario que es un dato alfabético y su contraseña que puede ser un dato alfa numérico. En la Figura 25 se muestra el formulario de autenticación y la Figura 26 muestra su respectivo diagrama de flujo.

Figura 25. Formulario de autenticación

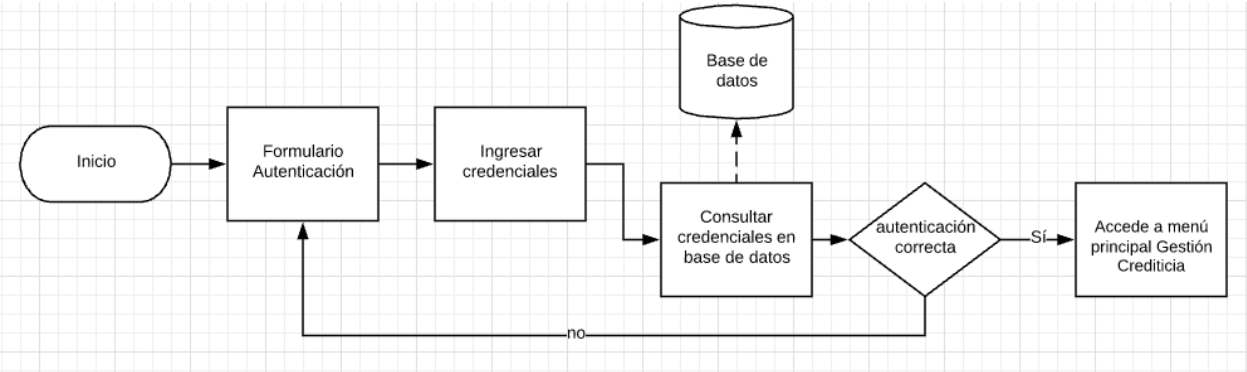


Figura 26. Diagrama de flujo autenticación

La Figura 27 muestra el menú principal de la aplicación que contiene dos módulos (Seguridad y Gestión Crediticia). Seguridad cuenta con dos formularios (Usuarios y Roles).

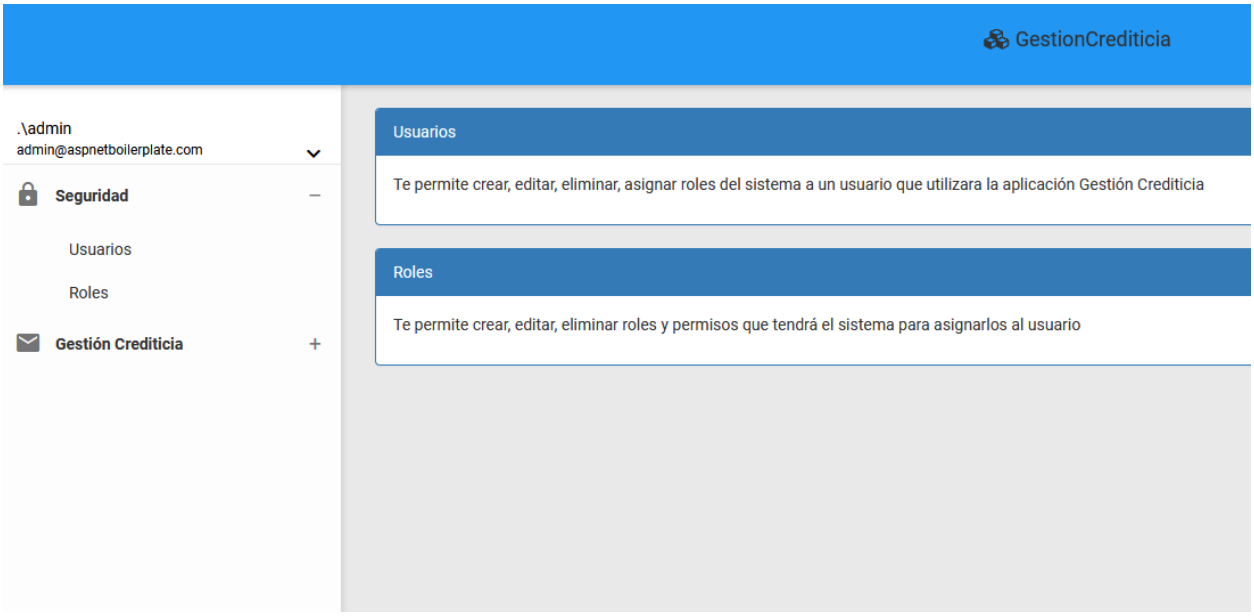


Figura 26. Menú mo[12]dulo Seguridad.

Para crear un usuario es necesario abrir el formulario Usuarios que desplegará una ventana con un listado de los usuarios creados los cuales se les puede eliminar, editar información básica y modificar sus roles. El formulario cuenta con un botón para la creación de un usuario que despliega una ventana modal para ingresar información básica y crear roles. La Figura 28 muestra el formulario Usuarios. Para crear un rol se abre el formulario roles donde se registra la información básica del rol y los permisos que éste contiene. La Figura 29 muestra el formulario de roles.

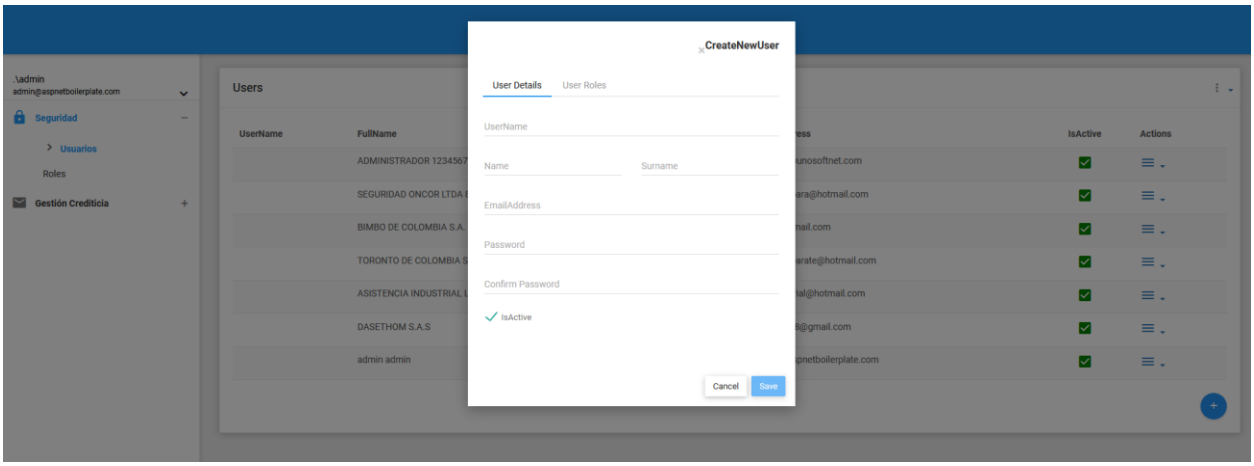


Figura 28. Formulario usuarios.

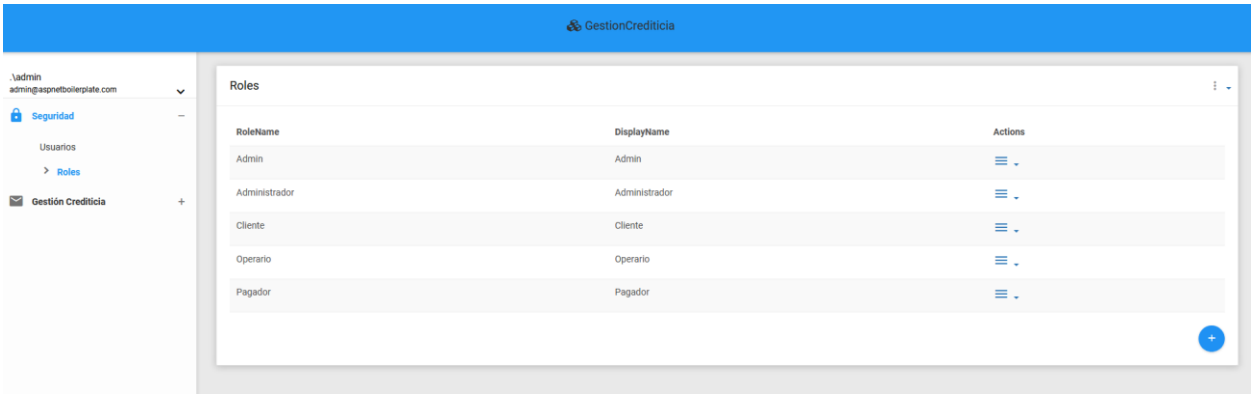


Figura 29. Formulario roles.

6.7 Módulo de gestión crediticia

El módulo de gestión crediticia cuenta con dos formularios (gestión y gestión masiva). El formulario gestión permite realizar a partir de los atributos descritos en la Tabla 6 una predicción de acuerdo al modelo seleccionado. La predicción queda almacenada en base de datos y puede ser consultada por parámetros como fecha de generación, id cliente y modelo predictivo. La Figura 30 muestra el formulario de gestión. La Figura 31 muestra diagrama de flujo del formulario gestión.

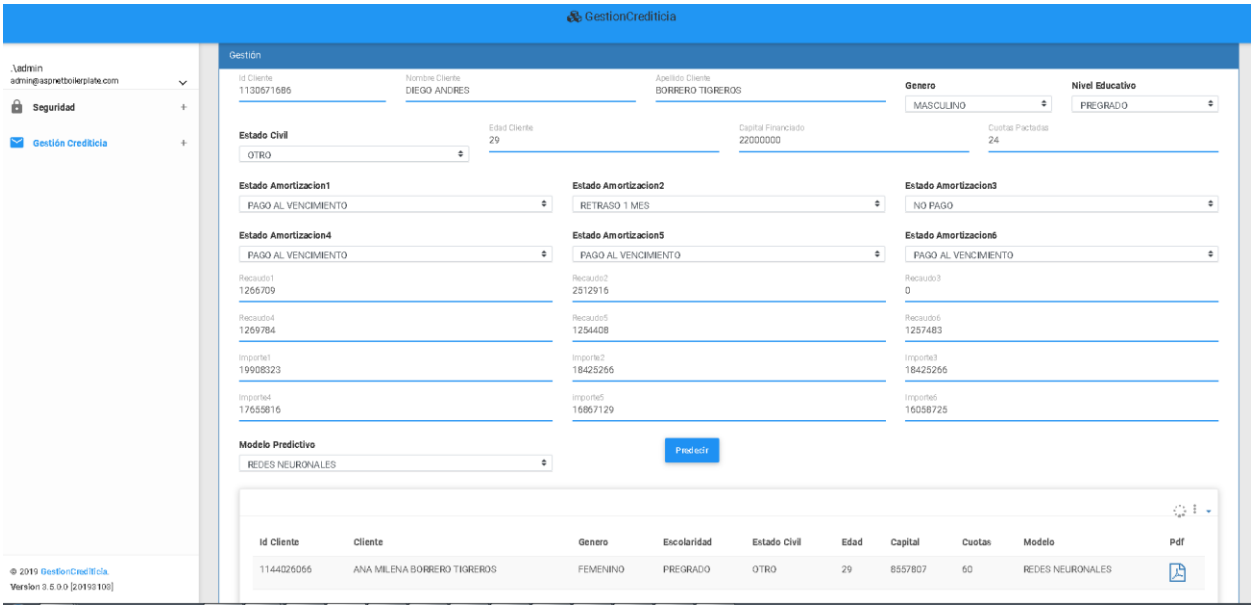


Figura 30. Formulario gestión.

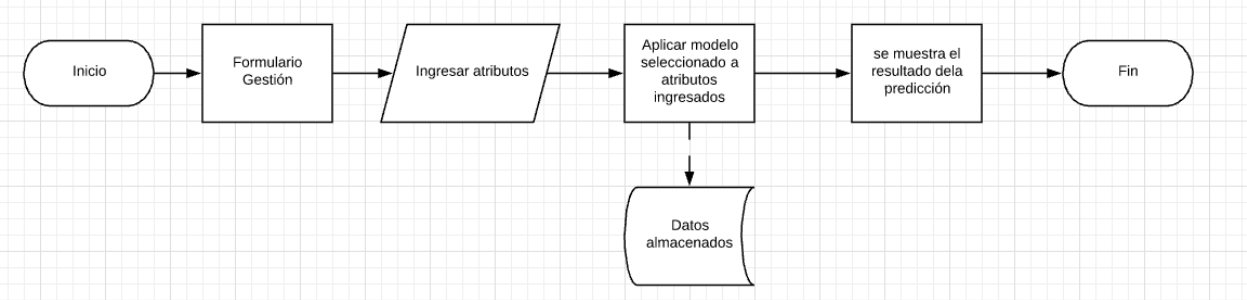


Figura 31. Diagrama de flujo Gestión

El formulario gestión masiva permite realizar predicciones a un conjunto masivo de atributos cargados en una plantilla en formato Excel que se debe adjuntar. La Figura 32 muestra el formulario gestiones masivas y la Figura 33 muestra el diagrama de flujo de gestiones masivas.

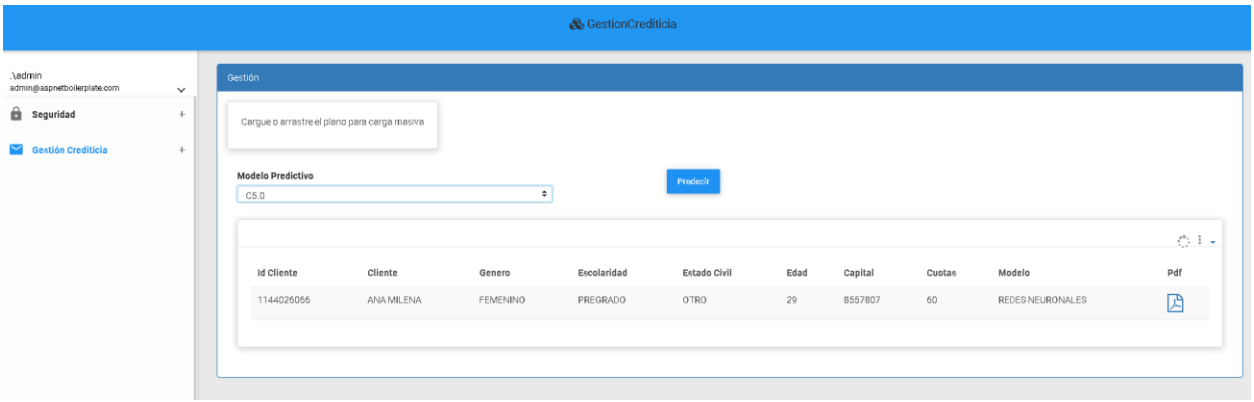


Figura 32. Formulario carga masiva

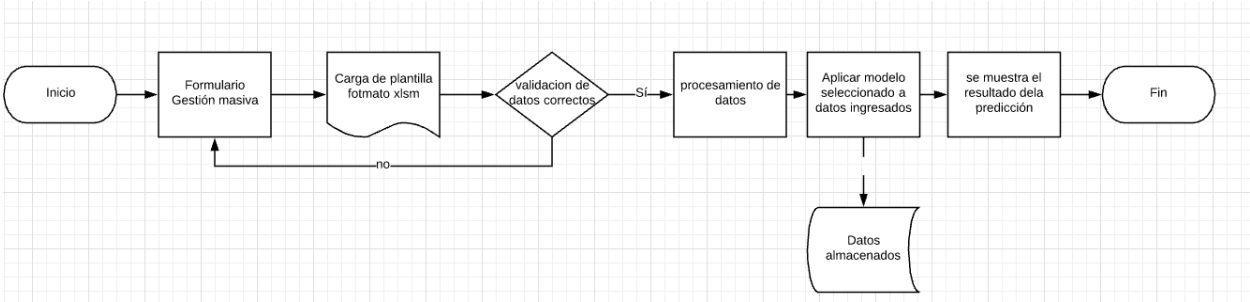


Figura 33. Diagrama de flujo Gestión masiva

6.8 Pruebas unitarias

Las pruebas unitarias permiten comprobar el código a nivel de módulos individuales con el fin de asegurar que funcionan correctamente por separado. Esto permite mejorar la calidad del software y disminuir en gran proporción la cantidad de fallos que se puede presentar. Para este trabajo de grado se generaron pruebas unitarias tipo TDD (Desarrollo dirigido por test) donde se generó una serie de pruebas iniciales básicas a partir de las historias de usuario generadas para el prototipo Gestión Crediticia. Por cada módulo se generó una serie de pruebas diseñadas en Jasmine que es una herramienta de testeo del AngularJS.

Por las historias de usuario para el módulo de seguridad se crearon 10 pruebas que tienen que ver con la creación de usuario, edición de usuario, eliminación de usuario, asignación de roles, creación de roles, edición de roles, eliminación de roles y asignación de permisos. Inicialmente el porcentaje de aciertos de estas pruebas fue del 0%. Una vez se inició con el desarrollo de los componentes en angular 4 las pruebas pasaron satisfactoriamente. Luego se realiza el proceso de refactorización donde se hace limpieza del código y se generan nuevas pruebas que tienen que ver con fallos críticos que se pueden presentar al ingresar datos que no corresponden al tipo de dato esperado o datos que pueden generar desbordamiento. En el proceso de refactorización se generaron 5 pruebas más y se implementaron en Jasmine.

Para el módulo de gestión crediticia se generaron 6 pruebas básicas a partir de las historias de usuario definidas que tienen que ver con la generación de una predicción a través de un conjunto de atributos obligatorios y la generación masiva de predicciones con una plantilla en Excel. Inicialmente el porcentaje de acierto de estas pruebas fue del 0% y luego de iniciar el desarrollo se alcanzó el 100%. Seguidamente se pasa al proceso de refactorización donde se limpia el código generado y se ejecutan nuevas pruebas que tienen que ver con los tipos de datos esperados en la plantilla para carga masiva con el fin de minimizar la cantidad de fallos que se pueden presentar en este módulo.

CAPÍTULO 7

Conclusiones y trabajo futuro

En este capítulo se presentan las conclusiones fundamentales que derivan de este trabajo de grado y se propone un trabajo futuro que permita ampliar y mejorar los resultados obtenidos.

7.1 Conclusiones

Para este trabajo de grado se propuso una aplicación prototipo que permita evaluar el riesgo crediticio a partir de un análisis predictivo sobre deudores actuales que pueden incurrir en mora en entidades financieras tipo pymes utilizando un conjunto de atributos que describen pagos anteriores e información básica de los clientes. Para esto se propuso diversos modelos utilizando técnicas de aprendizaje supervisado. Dado a los resultados obtenidos por las pruebas de evaluación aplicadas a cada modelo y comparado con los modelos aplicados en el estado del arte se puede decir que los algoritmos de árboles de decisión tuvieron un buen rendimiento ya que sus resultados están en el rango esperado. Con respecto a las redes neuronales los resultados no estuvieron en el rango esperado comparado con el estado del arte presentado en (Turkson et al., 2016) ya que presentan una diferencia de un 14% en la exactitud y el algoritmo que tuvo un bajo rendimiento que es el SVM tiene una diferencia de 17% comparado con los resultados obtenidos en el estado del arte. A continuación se presentan las conclusiones más importantes por cada modelo generado.

- **Redes Neuronales tipo perceptrón multicapa**. La red propuesta tiene el mejor promedio en sensibilidad que es el parámetro más importante para este problema ya que realizar una predicción que clasifique un cliente como un cliente que no va incumplir el pago de su cuota y termine incumpliendo es más costoso que un cliente que se clasifique como un cliente que incumplirá y termine cumpliendo con el pago. La sensibilidad alcanzada por la red propuesta en la sección 4.2.1 es de 71.83%. En otro parametro de medicion la red neuronal artificial obtuvo el cuarto puesto en promedio de exactitud utilizando validación cruzada con k=10 alcanzando un 63.62%. Esto implica que otros modelos generados en este trabajo de grado se ajustan mejor a los datos suministrados por la entidad financiera. De igual manera las redes neuronales obtuvieron el cuarto puesto en el promedio del área bajo la curva con un 69.91% que indica la probabilidad de que la clasificación realizada a un cliente que incumplirá el pago de su cuota sea más correcto que el de un cliente que no incumplirá el pago de su cuota escogida al azar. El análisis de sensibilidad indica que en este modelo las variables más importantes en correlación con la variable independiente son el Género y el Recaudo3.

- **Algoritmo C4.5** El árbol de decisión generado alcanza el segundo puesto en sensibilidad con un 64.28% como lo muestra la Tabla 13. El promedio de exactitud de este modelo es de 75.7% utilizando validación cruzada con k=10 y alcanza el segundo puesto lo que indica que el modelo tiene un mejor ajuste a los datos que las redes neuronales. En el área bajo la curva este modelo tiene un 79.73% considerado un resultado bueno que indica la efectividad al clasificar correctamente los clientes. El análisis de sensibilidad aplicado a este modelo arroja que las variables de mayor importancia son el recaudo6 y el EstadoAmortizacion6 que indican que estas variables son las que más aportan para la predicción que hace el modelo.

- **Algoritmo Random Forest.** El árbol de decisión generado alcanza el cuarto puesto en promedio de sensibilidad con un 57.14% pero obtiene el mejor promedio de exactitud con un 78.31% debido a que el modelo se especializó en detectar casos negativos. Esto se puede dar por el desequilibrio de las clases, algo característico en este problema ya que son menos los clientes que incumplen que los clientes que cumplen el pago de sus obligaciones. En el promedio de área bajo la curva se obtiene el mejor resultado lo que indica que el modelo clasifica los clientes de forma correcta alcanzando un valor de 88.29%. Las variables con mayor importancia con respecto a la variable dependiente son Recaudo6 y EstadoAmortizacion6 que indican que estas variables aportan mucho para la predicción del modelo y la de menor importancia es el Género.

- **Algoritmo C5.0** Este algoritmo alcanzó el tercer puesto en el parámetro sensibilidad con un 64.28%. De igual manera alcanza el tercer puesto en la exactitud promedio con un 64.93% que se calculó con validación cruzada de K=10. El modelo generado utilizando el algoritmo C5.0 obtuvo el segundo puesto en área bajo la curva con un resultado de 80.56% lo que indica que el modelo clasifica los clientes que incumpliran y los que cumpliran con el pago de su cuota. Además, el modelo cuenta con el hiperparámetro *cost* utilizado en casos donde hay un desequilibrio en los datos independientes con los que se genera el modelo. Esto implica que es una buena opción para trabajar con este tipo de problemas. El análisis de sensibilidad de este modelo arrojo que la variable independiente más importante para la predicción de la variable dependiente es el EstadoAmortizacion6 y la de menor importancia es el Recaudo3.

- **SVM (Máquinas de soporte vectorial)** Este algoritmo no tuvo un buen rendimiento ya que obtuvo el último lugar en todas las métricas utilizadas para evaluar los modelos.La sensibilidad alcanzada por este modelo es de 44.44% y su exactitud promedio utilizando la validación cruzada de k=10 es de 60.32%. En el área bajo la curva el resultado es de 59.30% lo que indica que el modelo tiende a realizar predicciones que se acerca a una estimación pseudoaleatoria. Aunque se utilizaron diversos kernel variando los hiperparámetros asociados a éstos, no se obtuvo una buena separación entre las dos clases impidiendo una clasificación correcta en las predicciones. Las variables de mayor importancia para este modelo es el EstadoAmortizacion6 y las CuotasPactadas y la variable EstadoAmortizacion5 presenta un valor cercano a 0 que indica que su aporte no es importante en la predicción de la variable dependiente.

7.2 Trabajo Futuro

La tesis presentada deja abierta la posibilidad de mejorar los resultados obtenidos mediante estrategias que permitan identificar si los atributos independientes obtenidos en este trabajo de grado son suficientes para generar modelos que permitan predecir riesgo crediticio o si se mejora reduciendo la cantidad de atributos ya que este problema hace parte de un sistema caótico. De igual manera a través de este trabajo de grado se puede plantear otro problema que consiste en poder predecir qué deudor es más susceptible en salir de una situación de mora para concentrar los esfuerzos de los agentes en el área de gestión crediticia sobre estos tipos de clientes.

8. Referencias

Portafolio. (2017) Bancos Colombianos han hecho la tarea en transformación digital Retrieved 08 19, 2017. Recuperado de http://www.portafolio.co/economia/banca-colombiana-lider-en-transformacion-digital-508861

Legislación. (2002). "Gestión del riesgo de crédito: modificaciones". Legislación, 100 (1188), 824-845.

Colprensa. (2017). Aumenta la lista de deudores morosos en el país. 03 20, 2017. Recuperado de http://www.elpais.com.co/economia/en-enero-el-valor-de-la-cartera-vencida-crecio-25-informo-la-superfinanciera.html.

Superintendencia Financiera. (2017) Conformación del Sistema financiero Colombiano Retrieved 10 15,2017. Recuperado de https://www.superfinanciera.gov.co/jsp/loader.jsf?lServicio=Publicaciones&lTipo=publicaciones&lFuncion=loadContenidoPublicacion&id=11268&dPrint=0

Turkson, Baagyere & Wenya (2016, 13 de octubre). A Machine Learning Approach for Predicting Bank Credit Worthiness. IEEE

Aboobyda & Tarig (2016). Developing Prediction Model of Loan Risk in Banks Using Data Mining. International Journal

Rogério et al (2016). Predicting Recovery of Credit Operations on a Brazilian Bank. IEEE

Ayala Villegas Sabino. (2005, Julio 12). Créditos financieros. Recuperado de https://www.gestiopolis.com/creditos-financieros/

César, S. (2004). Diccionario de términos económicos. Editorial Universitaria.

Caballero, F. (2012). Economipedia. Recuperado en http://economipedia.com/definiciones/solvencia.html

IBM Knowledge Center. (s.f). Recuperado en https://www.ibm.com/support/knowledgecenter/es/SSLVMB_22.0.0/com.ibm.spss.statistics.help/spss/base/idh_regs.htm

Eibe, F., Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Recuperado de https://www.cs.waikato.ac.nz/ml/weka/

Gentleman, R. (1997). The R Project for Statistical Computing. Recuperado de https://translate.google.com.co/translate?hl=es&sl=en&u=https://www.r-project.org/&prev=search

Inza, I. y Calvo, B. (s.f) h2o: big data analysis y c´omputo paralelo en R. Recuperado en http://www.sc.ehu.es/ccwbayes/members/inaki/tmp/Yvan-teaching-material/h2o-Tutorial.pdf

Pérez, J. (2013). Análisis ROC. Recuperado en https://estadisticaorquestainstrumento.wordpress.com/2013/02/13/tema-23-analisis-roc/

Carlos Arturo García (2016) Deudores bancarios empiezan a 'colgarse' con sus obligaciones. Retrieved 10 15, 2017. Recuperado de http://www.eltiempo.com/economia/sectores/deudas-sin-pagar-crecen-en-colombia-42539

I-Cheng Yeh (2016). UCI machine learning repository. Irvine, CA: University of California, School of Information and Computer Science Recuperado de http://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients#

Khashman, A. (2010). Neural networks for credit risk evaluation: Investigation of different neural models and learning schemes. Elsevier

Asunción, A., & Newman, D. J. (2007). UCI machine learning repository. Irvine, CA: University of California, School of Information and Computer Science. Recuperado de http://www.ics.uci.edu/~mlearn/MLRepository.html

Mitchell, T. (1997). Machine Learning. McGraw-Hill Science/Engineering/Math.

Pérez López, C. (2008). Minería de datos: técnicas y herramientas. Madrid.

McCrea, N. (2014). An Introduction to Machine Learning Theory and Its Applications: A Visual Tutorial with Examples. United States.

Matich Damián (2001). Redes Neuronales: conceptos básicos y aplicaciones. Rosario

Canós, Letelier & Penadés (2003, 12 de noviembre). Metodologías ágiles en desarrollo de software. Alicante-España.

Beck Kent (1999). Extreme Programming Explained. Embrace Change. Book Review.

Vapnik (1998). Statistical Learning Theory", Wiley, New York.

Hecht-Nielsen R. (1990) Neurocomputing, Addison-Wesley Publishing Company, Reading, Massachusetts.

Flórez & Fernández, (2008) Las Redes Neuronales Nrtificiales, Netbiblo.